

**INTEGRASI PANGKALAN DATA PERGURUAN TINGGI
DENGAN TEKNOLOGI WEB SERVICE**

BENNY LEONARD ENRICO PANGGABEAN

P2700211002



PROGRAM STUDI TEKNIK ELEKTRO

PROGRAM PASCASARJANA

UNIVERSITAS HASANUDDIN

MAKASSAR

2013

**INTEGRASI PANGKALAN DATA PERGURUAN TINGGI
DENGAN MENGGUNAKAN TEKNOLOGI WEB SERVICE**

Tesis

Sebagai salah satu syarat untuk mencapai Gelar Magister

Program Studi

Teknik Elektro

Disusun dan Diajukan oleh

BENNY LEONARD ENRICO PANGGABEAN

P2700211002

Kepada

PROGRAM STUDI TEKNIK ELEKTRO

PROGRAM PASCASARJANA

UNIVERSITAS HASANUDDIN

MAKASSAR

2013

TESIS

INTEGRASI PANGKALAN DATA PERGURUAN TINGGI

DENGAN MENGGUNAKAN TEKNOLOGI WEB SERVICE

Disusun dan diajukan oleh

Benny Leonard Enrico Panggabean

Nomor Pokok P2700211002

Telah dipertahankan di depan Panitia Ujian Tesis

Pada Tanggal 19 Agustus

Dan dinyatakan telah memenuhi syarat

Menyetujui

Komisi Penasihat

Dr.Armin Lawi, S.Si, M.Eng
Ketua
Ketua Program Studi Teknik elektro

Amil Ahmad Ilham, ST, MIT, Ph.D
Sekretaris
Direktur Program Pascasarjana
Universitas Hasanuddin

Prof.Dr.Ir.H.Salama Manjang, MT

Prof.Dr.Ir.Mursalim, MT

PERNYATAAN KEASLIAN THESIS

Yang bertanda tangan dibawah ini

Nama : Benny Leonard Enrico Panggabean

Nim : P2700211002

Program Studi : Teknik Elektro

Konsentrasi : Teknik Informatika

Menyatakan dengan sebenarnya bahwa tesis yang saya tulis ini benar – benar merupakan hasil karya saya sendiri, bukan merupakan pengambil-alihan tulisan atau pemikiran orang lain. Adapun kutipan atau rujukan sebagai sumber informasi yang saya gunakan dari penulis lain, telah saya sebutkan namanya pada daftar pustaka tesis ini.

Apabila dikemudian hari terbukti bahwa tesis adalah hasil karya orang lain maka saya bersedia menerima sanksi apapun sesuai peraturan yang berlaku.

Makassar, September 2013

Penulis

ABSTRAK

Benny Leonard Enrico Panggabean. Integrasi Pangkalan Data Perguruan Tinggi Dengan Teknologi Web Service (dibimbing oleh **Dr. Armin Lawi, M.Eng, S.Si** dan **Amil Ahmad Ilham, ST, MIT, Ph.D**)

Pangkalan Data Perguruan Tinggi (PDPT) merupakan rekaman data menyangkut penyelenggaraan Perguruan Tinggi baik akademik maupun non akademik. PDPT bertujuan sebagai layanan bagi setiap Perguruan Tinggi untuk dapat memperpanjang ijin penyelenggaraan Program Studi, menyediakan pusat penyimpanan data pelaporan akademik dan non akademik Perguruan Tinggi, mendukung Sistem Penjaminan Mutu Internal (SPMI), Sistem Penjaminan Mutu Eksternal (SPME), dan mendukung kebutuhan *benchmarking* Perguruan Tinggi, mendukung sistem berskala *enterprise*, menjamin keamanan data dilevel jaringan dan basisdata, dan mendukung integritas dan konsistensi data.

Sejalan dengan tujuan PDPT bagi DIKTI, dimana Ditjen DIKTI memerlukan data dan informasi yang akurat berkaitan dengan pendidikan tinggi dalam rangka menentukan kebijakan-kebijakan, diperlukan integrasi data dan informasi (PT, kopertis, BAN-PT, dan lain lain), sehingga Ditjen DIKTI dapat mengevaluasi standar dan kebijakan yang telah dikeluarkan berdasarkan data dan informasi yang dikumpulkan dan menjadi dasar analisis untuk pembuatan kebijakan – kebijakan Ditjen DIKTI berikutnya.

Sinkronisasi dan integrasi dalam proses pengiriman data dari Program Studi ke PDPT hingga Ditjen DIKTI, digunakan perangkat lunak dengan teknologi Web Service Push/ ESBED yang mendukung interoperabilitas dan interaksi data antar mesin internal PT dengan mesin DIKTI menggunakan format data XML. Layanan ini akan membuat mesin internal PT mengirimkan data ke mesin DIKTI secara aktif.

Kata Kunci : PDPT, EPSBED, Web-Service.

ABSTRACT

Benny Leonard Enrico Panggabean. University database integration using web service technology (guided by **Dr. Armin Lawi, M.Eng, S.Si** and **Amil Ahmad Ilham, ST, MIT, Ph.D**)

University database are data records about university management both in academic and non-academic activity. University database purpose are to be a service for every university to extend the license of course concentration departments, provide data storage center for academic and non-academic activity, support both external and internal quality assurance, and support for benchmarking requirement of universities, support for enterprise scale system, ensure data security both in database and network level, and support for data integrity and consistency.

In line with the objective university database for DIKTI, where DIKTI need for accurate data and information related in higher education for determine policy, a need for data and information integration from universities, national accreditation council, and others so that DIKTI can evaluate standard and policy which has been issued based on gathered data and information so it can be basic analysis for policy making DIKTI in the feature.

Software with capability to send data using web services is required to synchronization and integration of data submission process from course concentration department to university database and university database to DIKTI, which support for interoperability and data interaction from university server to DIKTI server which is communicating using XML format data. This service will make university server can actively send data to DIKTI server.

Key Words: university database, EPSBED, Web-Service

KATA PENGANTAR



Assalamu 'alaikum Wr. Wb.

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat dan karunianya, shalawat serta salam semoga tercurah kepada Rasulullah SAW, sehingga penulis dapat menyelesaikan Thesis yang berjudul **“INTEGRASI PANGKALAN DATA PERGURUAN TINGGI DENGAN TEKNOLOGI WEB SERVICE”**.

Adapun tujuan dari penyusunan thesis ini adalah untuk memenuhi salah satu syarat dalam menyelesaikan studi jenjang strata dua (S-2) di Program Studi Teknik Elektro - Teknik Informatika, Universitas Hasanuddin.

Dengan keterbatasan ilmu dan pengetahuan yang dimiliki oleh penulis, maka penulis membutuhkan peran serta dari pihak lain dalam proses penyelesaian thesis ini. Oleh karena itu ijinilah penulis untuk menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Ayahanda Alm.Theo Mathias Panggabean, ibunda Hj.Sukarni Musa, dan Alpha Leo Erwin Panggabean yang telah memberikan kesempatan untuk kuliah lagi di Universitas Hasanuddin, doa dan dorongan moril maupun materi yang tiada henti.

2. Bapak Prof. Dr. dr. Idrus A Paturusi., Selaku Rektor Universitas Hasanuddin.
3. Bapak Prof. Dr.Ing. Ir. Wahyu H. Piarah, MSME., Selaku Dekan Fakultas Teknik UNHAS.
4. Bapak Prof. DR. Ir. H. Salama Manjang, M.T., Selaku Ketua Program Studi Pasca Sarjana Teknik Elektro UNHAS.
5. Bapak Dr. Armin Lawi, M.Eng, S.Si, Selaku Dosen Pembimbing I dan Bapak Amil Ahmad Ilham, ST, MIT, Ph.D., selaku Dosen Pembimbing II. Terima kasih karena telah banyak meluangkan waktu untuk memberikan bimbingan, saran dan nasehatnya selama penyusunan thesis ini.
6. Bapak Muhammad Niswar, ST., MIT., Ph.D , Bapak Dr.Ir.H.Andani Achmad, MT, dan Dr.Ir.Zulfajri B.Hasanuddin, M.Eng yang menguji thesis ini.
7. Jajaran Dosen dan Staff Fakultas Teknik, Pasca Sarjana Teknik Elektro - Teknik Informatika UNHAS.
8. Poetri Lokapitasari Belluano, Muhammad Luthfy Kasim, Ucok Sinaga, Gafur Jamada, Abdul Mubarak, Adam M Tanniewa Adnan Nur dan teman - teman di Teknik Elektro -Teknik Informatika 2011, terima kasih atas perhatian dan bantuannya.
9. Bapak Irham Nusally Regional Sales & Operations Manager AFM di Cameron Aftermarket, V&M Division dan Bapak Erwin Tejakusuma Operation Manager at Geographe Energy,

Cameron V&M Aftermarket yang memberikan pengetahuan tentang etos dan prosedur dalam bekerja serta teman – teman di Geographe Energy.

10. Bapak Andri Heryandi, S.T, M.T, Bapak Irawan Afrianto , S.T, M.T, Bapak Eko Budi Setiawan, S.Kom., M.T, Ibu Dian Dharmayanti, S.T.,M.Kom., Ibu Tati Harihayati M., S.T.,M.T, Irfan Maliki S.T, M.T dan teman – teman IF-1 2010 dan IF-3 2011 Teknik Informatika Universitas Komputer Indonesia
11. Muhammad Ilham Syahrul, Yusri Affandy, Yushar Mustafa, Cristian Ade Chandra dan Teman – teman SMA Negeri 1 Makassar angkatan 2000
12. Kiki Maulana Adhinugraha, S.Kom.,M.T Institut Teknologi Telkom Indonesia dan Aditya Gunawan Syamsudin.
13. Ibu Rosa de Lima E.P., Dra, M.T, Bapak Henry Pantas Panggabean, Kim Siung, Purnawan Xiang, Husein Syurdi, Hady Tandibali, Eben Heizer Katilik dan teman – teman di fakultas MIPA Ilmu Komputer Universitas Katolik Parahyangan Angkatan 2001
14. Maria Christine, Niko Ferdian, Niko Lukman, dan Teman – teman di bukit sastra 111
15. Bapak Ardha TP, Fandy Ferdian, M Fadlil Sangaji, Roy Victor Sihombing, Leo C Sinaga, Desanto Wuryantoro, Mangestya W Argyaputri, Rama Ashryputro, Sh Wibowo, dan Novelya Marta

Uly Napitupulu, Agung Rahmat Saleh dan teman – teman di Corporate Information System and Technology Division Samudra Indonesia.

16. Bapak Syahrul S.Kom, M.Si., Teman – Teman kelas AK-10 Fakultas Ilmu Komputer Program Studi Teknik Informatika, dan teman – teman kelas B-10, C-10, 1-10 dan 3-10 di Fakultas Keperawatan Program Studi Kebidanan Universitas Indonesia Timur.
17. Komunitas CakePHP dan Komunitas CakePHP Indonesia
18. Serta semua pihak yang telah turut membantu dalam penyusunan thesis ini, yang tidak bisa di sebutkan semuanya satu persatu.

Dalam penulisan thesis ini, penulis telah berusaha semaksimal mungkin, walaupun demikian penulis menyadari bahwa karya tulis ini jauh dari kesempurnaan yang diharapkan. Oleh karena itu penulis akan selalu menerima segala masukan yang bersifat membangun sehingga dapat menyempurnakan thesis ini.

Akhir kata, penulis berharap semoga thesis ini dapat bermanfaat bagi penulis pada khususnya dan pembaca pada umumnya.

Wassalamu 'alaikum Wr .Wb.

Makassar, September 2013

Penulis

DAFTAR ISI

ABSTRAK	i
KATA PENGANTAR	iii
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR SIMBOL	xv
BAB I PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Rumusan Masalah	5
C. Tujuan Penelitian	6
D. Kegunaan Penelitian	6
E. Ruang Lingkup / Batasan Penelitian	7
BAB II TINJAUAN PUSTAKA.....	8
A. Model Paradigma Perancangan Sistem	8
1. Design Pattern.....	8
2. Convention Over Configuration	8
3. Model View Controller	9
B. Data Center	11
C. Basis Data.....	12
1. Tujuan Basis Data	15
2. Manfaat / Kelebihan Basis Data	18

D. Basis Data Terdistribusi.....	21
1. Tipe Basis Data Terdistribusi.....	22
2. Arsitektur Basis Data Terdistribusi	23
3. Penyimpanan data pada seistem terdistribusi.....	25
E. Web Service.....	25
1. Arsitektur Web Service	26
2. Operasi – Operasi Web Service	27
3. Komponen – Komponen Web Service	28
F. XML (Extensible Markup Language).....	29
G. Simple Object Access Protocol (SOAP)	29
H. Integrasi Data	35
1. <i>Enterprise Application Integration</i> (EAI)	36
2. <i>Service-Oriented Architecture</i> (SOA)	36
3. <i>Federation</i>	37
4. <i>Extract, Transform, Load</i> (ETL).....	37
I. Dasar hukum PDPT.....	38
J. Penelitian Terkait.....	38
1. <i>Information Integration Using Logical Views</i> (1997).....	39
2. Data Integration against Multiple Evolving Autonomous Schemata (2001)	39
3. A Practical Approach To Merging MDDM.....	39
1. The Implications and Impacts of Web Services to Electronic Commerce Research and Practices	40

2. A Survey on Web Service Composition.....	40
3. Web Service Selection Based On Ranking of QoS Using Associative Classification	40
K. Kerangka konseptual	42
BAB III METODE PENELITIAN	44
A. Jenis dan Sumber Data	44
1. Data Primer	44
2. Data Sekunder.....	44
B. Analisis Data.....	45
C. Perancangan Sistem.....	45
1. Use Case Diagram	46
2. <i>Class</i> Diagram	47
3. <i>Sequence</i> Diagram.....	54
D. Jenis Penelitian	68
E. Instrumen Pengumpul Data	68
1. Perangkat Keras.....	68
2. Perangkat Lunak	68
F. Tempat dan Waktu Penelitian.....	68
G. Tahapan Penelitian.....	69
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	71
A. HASIL PENELITIAN	71
1. Sistem PDPT-Universitas	72

2. Komponen form sebagai alat bantu pada proses memasukkan data.....	77
3. Validasi Internal Sistem	79
4. Pengujian Sistem dengan metode <i>whitebox</i>	82
5. Pengujian Sistem dengan metode <i>blackbox</i>	84
6. Pengujian fungsional webservice PDPT-DIKTI	86
7. Simulator PDPT-DIKTI	88
B. PEMBAHASAN.....	90
BAB V KESIMPULAN DAN SARAN	105
A. Kesimpulan	105
B. Saran	105
DAFTAR PUSTAKA.....	cvi

DAFTAR GAMBAR


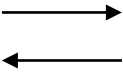



Gambar 1.1 Gambaran Umum Sistem Yang Di Usulkan.....	5
Gambar 2.1 Arsitektur Model View Controller (sumber cakephp)	10
Gambar 2.2: Basis Data Heterogen.....	22
Gambar 2.3: Sistem Client Server	23
Gambar 2.4: Arsitektur Web Service	27
Gambar 2.5: Komponen Web Service	28
Gambar 2.6: Contoh Format Message SOAP	34
Gambar 3.1: Use Case Diagram Sistem.....	46
Gambar 3.2: Class Diagram Sistem untuk entitas utama.....	48
Gambar 3.3: <i>Sequence</i> Diagram pengiriman data mahasiswa	54
Gambar 3.4: <i>Sequence</i> Diagram pengiriman data matakuliah	55
Gambar 3.5: <i>Sequence</i> Diagram pengiriman data mahasiswa	57
Gambar 3.5: <i>Sequence</i> Diagram pengiriman data mengajar dosen	58
Gambar 3.6: <i>Sequence</i> Diagram pengiriman data transaksi riwayat status mahasiswa.....	59
Gambar 3.7: <i>Sequence</i> Diagram pengiriman data transaksi nilai mahasiswa.....	61
Gambar 3.8: Activity diagram pengiriman data mahasiswa.....	62

Gambar 3.9: Activity diagram pengiriman data matakuliah	63
Gambar 3.10: Activity diagram pengiriman data transaksi aktivitas kuliah mahasiswa	64
Gambar 3.11: Activity diagram pengiriman data transaksi aktivitas mengajar dosen	65
Gambar 3.12: Activity diagram pengiriman data transaksi riwayat status mahasiswa	66
Gambar 3.13: Activity diagram pengiriman transaksi nilai mahasiswa	67
Gambar 3.14: Alur Tahapan Penelitian	69
Gambar 4.1 : Arsitektur pengembangan system antara server PDPT-Universitas dengan PDPT-DIKTI	71
Gambar 4.2 : Chart data invalid internal	73
Gambar 4.3 : chart data yang belum terverifikasi	74
Gambar 4.4 : chart data yang invalid	74
Gambar 4.5 : chart data yang terverifikasi	75
Gambar 4.6 : interface untuk mengirim data Prodi ke PDPT-DIKTI	76
Gambar 4.7 : interface hasil respon data layanan web service PTPT-DIKTI.....	76
gambar 4.8: komponen autocomplete pada modul transaksi aktivitas mengajar dosen	77


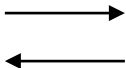

Gambar 4.9: komponen calendar pada modul matakuliah	78
Gambar 4.10: Form input data pada modul matakuliah	80
Gambar 4.11: Validasi <i>Client Side</i> Form input pada modul matakuliah ...	81
Gambar 4.12: Validasi <i>Server Side</i> pada form transaksi aktivitas mengajar dosen	82
Gambar 4.13 : Skenario Pengujian whitebox	83
Gambar 4.14 : code hasil dari pengujian whitebox.....	83
Gambar 4.15 : Pengujian modul master mahasiswa dengan metode whitebox.....	84
Gambar 4.16 : Penggunaan memori pada pengujian blackbox.....	85
Gambar 4.17 : Penggunaan memori pada pengujian blackbox.....	86
Gambar 4.18 : Pengujian Access web service PDPT-DIKTI	87
Gambar 4.19 : Pengujian Response web service PDPT-DIKTI.....	87
Gambar 4.20: Modul Master Mahasiswa pada simulator	88
Gambar 4.21: Modul Transaksi Aktivitas Mengajar pada simulator	88
Gambar 4.22: Modul Master Mahasiswa pada simulator setelah memvalidasi data yang dikirim oleh system	89
Gambar 4.23: Modul Transaksi Aktivitas Mengajar pada simulator setelah memvalidasi data yang dikirim oleh system	89

DAFTAR SIMBOL


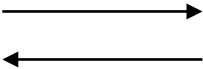

Simbol Sekuens Diagram

Simbol	Nama	Keterangan
	Aktor	Merupakan pelaku yang berinteraksi dengan sistem
	Aliran	Menunjukkan data – data yang mengalir pada sistem
	Boundary Object	Boundary objects, merepresentasikan antar muka antara aktor dan sistem. Dalam sebuah aplikasi web, biasanya berbentuk halaman web
	Entity Object	Entity objects, adalah sesuatu yang tetap ada setelah kejadian-kejadian pada use case
	Control Object	Control objects, hal ini merepresentasikan proses

Simbol Use Case Activity Diagram

SIMBOL	NAMA	KETERANGAN
	Initial State	Merupakan awal dari suatu aktifitas
	Aliran	Menunjukkan data – data yang mengalir pada sistem
	Final State	Merupakan akhir dari suatu aktifitas
	State	Merupakan sebuah aktifitas yang dilakukan
	Decision	Merupakan simbol pilihan aktifitas

Simbol Use Case Diagram

SIMBOL	NAMA	KETERANGAN
	aktor	Merupakan pelaku yang berinteraksi dengan sistem
	Aliran	Menunjukkan data – data yang mengalir pada sistem
	Proses	Menunjukkan data – data yang mengalir pada sistem

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Pangkalan Data Perguruan Tinggi (PDPT) adalah rekaman data menyangkut penyelenggaraan Perguruan Tinggi baik akademik maupun non akademik, dimana PDPT memerlukan pengelolaan tersistem agar dapat menghasilkan informasi yang bermakna, data perguruan tinggi digunakan untuk mendukung pengelolaan perguruan tinggi (mencakup semua siklus manajemen), validitas dan kelengkapan data menjadi tanggung jawab bersama antara Ditjen DIKTI dan Perguruan Tinggi.

PDPT bertujuan untuk mendefinisikan dan mencari kesepadanan data yang ada diinternal Ditjen DIKTI dan entitas lainnya di siklus manajemen pendidikan tinggi, seperti BAN-PT, Kopertis, dan PSP (Pusat Statistik Pendidikan) Balitbang, menjamin integritas dan konsistensi antara data yang berasal dari ditjen DIKTI maupun entitas lainnya di siklus manajemen pendidikan tinggi, mendefinisikan dan menklarifikasi proses bisnis masing masing entitas didalam siklus manajemen pendidikan tinggi serta menjamin aliran data yang komprehensif dari dan Ditjen DIKTI dan entitas pendidikan tinggi lainnya, merancang dan mengimplementasikan database terpusat Ditjen DIKTI, serta menghasilkan informasi yang komprehensif serta menjamin integritas, konsistensi dan validitas data yang pada umumnya berasal dari database Ditjen DIKTI yang memiliki

struktur, platform, teknologi dan produk database yang berbeda.

Adapun manfaat PDPT adalah mendukung kebutuhan Perguruan Tinggi dalam memperpanjang ijin penyelenggaraan Program Studi, menyediakan pusat penyimpanan data pelaporan akademik dan non akademik Perguruan Tinggi, mendukung Sistem Penjaminan Mutu Internal (SPMI) dan Sistem Penjaminan Mutu Eksternal (SPME), mendukung kebutuhan *benchmarking* Perguruan Tinggi, mendukung sistem berskala *enterprise*, menjamin keamanan data dilevel jaringan dan basisdata, dan mendukung integritas dan konsistensi data. Sejalan dengan tujuan PDPT bagi DIKTI, dimana Ditjen DIKTI memerlukan data dan informasi yang akurat berkaitan dengan pendidikan tinggi dalam rangka menentukan kebijakan-kebijakan dan diperlukan integrasi data dan informasi (PT, kopertis, BAN-PT, dan lain lain), sehingga Ditjen DIKTI dapat mengevaluasi standar dan kebijakan yang telah dikeluarkan berdasarkan data dan informasi tersebut yang menjadi dasar analisis untuk pembuatan kebijakan-kebijakan Pendidikan Tinggi.

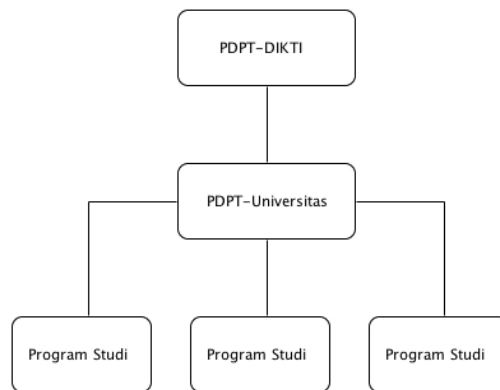
Fitur fitur dari PDPT adalah *Web Data Entry* (perangkat lunak yang mendukung pengiriman data ke mesin PDPT DIKTI menggunakan form berbasis web layout), *Web Loader* (perangkat lunak yang mendukung pengiriman data ke mesin PDPT DIKTI menggunakan file spreadsheet / excel dengan format yang telah ditentukan), *Web Service Push PDPT* (perangkat lunak yang mendukung interoperabilitas dan interaksi data antar mesin internal PT dengan mesin DIKTI menggunakan format data

XML. Layanan ini akan membuat mesin internal PT mengirimkan data ke mesin DIKTI secara otomatis, aktif dan terjadwal).

Bagi Perguruan Tinggi yang sudah mempunyai sistem informasi sendiri disarankan menggunakan *web service* dengan pencocokan atribut basis data Sistem Informasi Akademik Perguruan Tinggi dengan kamus data PDPT-DIKTI, dilain pihak bagi Perguruan Tinggi yang belum mempunyai sistem informasi agar dapat mengembangkan sistem informasi sesuai dengan kebutuhan masing – masing Perguruan Tinggi dan tetap mengacu pada dokumen kamus data PDPT-DIKTI, mengingat perangkat lunak masih terus ditingkatkan keterandalan dan kapasitasnya, tidak menutup kemungkinan terdapat perbedaan tampilan atau bentuk fasilitas menu yang belum tersedia, untuk itu aplikasi akan terus disempurnakan khususnya bila terdapat perubahan yang sangat substantif.

Masalah yang dihadapi pada saat ini adalah bagi Perguruan Tinggi yang belum memiliki sistem informasi tidak dapat memanfaatkan fasilitas *web service* yang disediakan oleh DIKTI, sehingga disarankan untuk membuat sistem informasi baru yang sesuai dengan kebutuhan Perguruan Tinggi dan mengacu pada kamus data PDPT-DIKTI, bagi Perguruan Tinggi yang sudah memiliki sistem informasi masalah yang dihadapi adalah bagaimana memetakan atribut data sistem informasi yang dimiliki dengan atribut kamus data PDPT-DIKTI.

Studi kasus dari penelitian ini bertempat di Fakultas Teknik Program S1 Teknik Informatika Universitas Hasanuddin Makassar sebagai *sample* program studi dan PDPT Universitas Hasanuddin Makassar, Program Studi Teknik Informatika masih menyimpan data dalam bentuk *spreadsheet* dan *word processor* untuk semua pelaksanaan kegiatan akademik dan merekapitulasi data tersebut kedalam aplikasi Evaluasi Program Studi Berbasis Evaluasi Diri (EPSBED) secara manual, sehingga proses pelaporan data kegiatan akademik dari program studi Program Studi Teknik Informatika ke PDPT-PT sangat memakan waktu, disisi lain jumlah operator yang mengetahui cara menggunakan aplikasi EPSBED di Program Studi Teknik Informatika sangat kurang, hal ini sangat menyulitkan bagi pengelola program studi untuk melakukan kegiatan penjaminan mutu internal maupun eksternal, dari sisi PDPT-PT hal ini sangat menyulitkan karena dapat menghambat kinerja kegiatan pelaporan ke PDPT-DIKTI.



Gambar 1.1 Gambaran Umum Sistem Yang Di Usulkan

Berdasarkan pemaparan latar belakang maka penulis berencana melakukan penelitian dengan judul **“Integrasi Pangkalan Data Perguruan Tinggi Dengan Teknologi Web Service”**, dimana dengan adanya sistem ini diharapkan dapat membantu proses pemusatan data, ketersediaan dan pemanfaatan data secara berkelanjutan.

B. Rumusan Masalah

- a. Bagaimana merancang dan membangun perangkat lunak sistem informasi ditingkat universitas yang memiliki integritas dan kompetensi data dengan PDPT-DIKTI.
- b. Bagaimana mengimplementasikan teknologi Web Service untuk melakukan integrasi basis data untuk layanan kebutuhan eksternal dimana sistem informasi PDPT-PT dapat mengirim data ke mesin internal PDPT-DIKTI secara aktif.

C. Tujuan Penelitian

Dengan memperhatikan latar belakang dan rumusan masalah diatas maka tujuan dari penelitian ini adalah

- a. Membangun Sistem Informasi yang memiliki integritas dan kompetensi data antar PDPT-PT dan PDPT-DIKTI.
- b. Membangun layanan dengan web service sehingga sistem PDPT-PT dapat mengirim data ke PDPT-DIKTI secara aktif serta berfungsi melakukan replikasi basis data.

D. Kegunaan Penelitian

Hasil penelitian ini dapat diharapkan berguna untuk hal – hal sebagai berikut :

- a. Memudahkan pengelolaan dan pemanfaatan data dari masing masing program studi di Universitas Hasanuddin Makassar.
- b. Memudahkan integrasi data dan akses informasi dari Program Studi ke PDPT-PT, dan integrasi data dan akses informasi dari PDPT-DIKTI.
- c. Memudahkan pihak eksekutif pada program studi Perguruan Tinggi untuk mengakses data dan informasi (kebutuhan internal).

Memudahkan pihak eksekutif diluar Program Studi untuk mengakses data dan informasi (kebutuhan eksternal).

E. Ruang Lingkup / Batasan Penelitian

Ruang Lingkup dari penelitian ini adalah

- a. Yang dimaksud dengan integrasi pada penelitian ini adalah pengkombinasian data yang berasal dari sumber basis data yang berbeda, dimana hasil dari kombinasi ini adalah penyediaan data secara keseluruhan kepada pengguna.
- b. Teknologi *web – service* yang digunakan adalah teknologi *Simple Object Access Protocol (SOAP)* versi 1.1.
- c. Sistem informasi yang dibuat adalah untuk Program Studi Teknik Informatika Fakultas Teknik Universitas Hasanuddin Makassar dan PDPT-PT Universitas Hasanuddin Makassar.

Diasumsikan bahwa basis data yang akan diintegrasikan adalah basis data yang baru.

BAB II

TINJAUAN PUSTAKA

A. Model Paradigma Perancangan Sistem

1. Design Pattern

Design Pattern (pola perancangan) adalah deskripsi dari object dan class yang saling berkomunikasi, dimana object dan class tersebut telah di modifikasi (d disesuaikan) untuk memecahkan permasalahan dalam perancangan (desain) secara umum dalam konteks tertentu (Erich Gamma,1998).

Pemberian nama, abstracts, dan pengidentifikasian adalah aspek kunci dari struktur perancangan umum yang berguna untuk membuat perancangan berorientasi objek yang dapat dipergunakan secara berulang – ulang.

2. Convention Over Configuration

Convention Over Configuration atau disebut juga *Coding By Convention* adalah sebuah paradigma perancangan perangkat lunak yang bertujuan untuk mengurangi jumlah keputusan yang harus diambil oleh pengembang perangkat lunak, untuk kesederhanaan tanpa harus kehilangan fleksibilitas (Russ Olsen, 2008:314).

3. Model View Controller

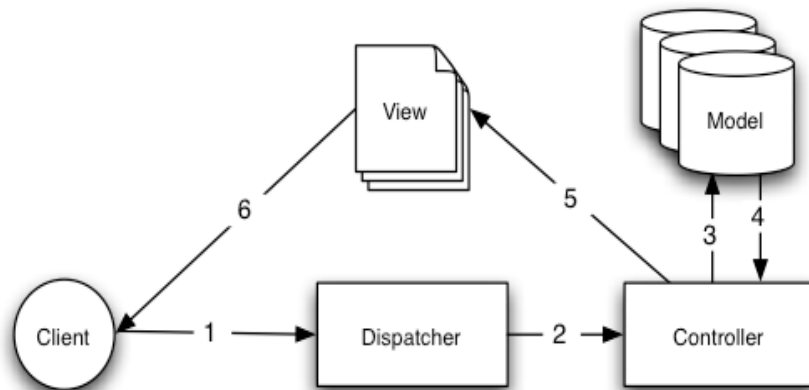
MVC (Model View Controller) merupakan sebuah metode untuk mengorganisasikan sebuah website dinamis (David Golding, 2008:3).

Dengan penjelasan tiap bagian sebagai berikut :

- a. Models adalah object yang merupakan representasi dari data pokok.
- b. View menampilkan keadaan dari model. Komponen ini bertanggungjawab untuk menampilkan informasi pada pengguna akhir.
- c. Controller menawarkan untuk mengubah keadaan model. Controller juga bertugas untuk mencari informasi dari model dan menyediakan data dinamis untuk views.

Hasilnya ketika terjadi permintaan (request) dari user / pengguna dilakukan proses dengan urutan sebagai berikut :

- a. Client melakukan proses request terhadap Controller melalui dispatcher.
- b. Dispatcher menunjuk controller yang sesuai dengan request client
- c. Controller merespon dengan meminta data dari Model.
- d. Model memberikan data yang diminta oleh controller.
- e. Controller mengolah tampilan dan memberikannya pada View.
- f. View mengembalikan tampilannya kepada pengguna yang meminta layanan.



Gambar 2.1 Arsitektur Model View Controller (sumber cakephp)

Pemisahan aplikasi menjadi tiga buah komponen memberikan beberapa kelebihan diantaranya adalah :

- a. Meningkatkan skalabilitas (kemampuan aplikasi untuk berkembang), jika aplikasi menjadi lambat disebabkan karena lambatnya akses database, sebagai contoh, anda dapat meng upgrade hardware yang menjalankan database tanpa ada efek terhadap komponen lainnya.
- b. Membuat perawatan lebih mudah, karena tiap komponen memiliki ketergantungan rendah pada komponen lainnya, membuat perubahan terhadap suatu bagian (baik memperbaiki atau menambah fungsionalitas), tidak menimbulkan efek terhadap komponen lain.
- c. Membuat aplikasi mudah dipindahkan, pemisahan yang jelas dari kode tiap komponen berarti tiap bagian dapat ditaruh dimesin terpisah jika diperlukan.

B. Data Center

Pusat data (Bahasa Inggris: data center) adalah suatu fasilitas yang digunakan untuk menempatkan sistem komputer dan komponen-komponen terkaitnya, seperti sistem telekomunikasi dan penyimpanan data. Fasilitas ini biasanya mencakup juga catudaya redundan atau cadangan, koneksi komunikasi data redundan, pengontrol lingkungan (mis. AC, ventilasi), pencegah bahaya kebakaran, serta piranti keamanan fisik (Maurico Arregoces, 2004:5).

Data center server adalah Pusat pemrosesan data yang didukung dengan perangkat pengolahan data tersebut. Disebut juga dengan pusat komputerisasi. Data center server merupakan server data terpusat dari jaringan di suatu jaringan, baik dalam jaringan lokal ataupun global, jaringan instansi ataupun perusahaan. Data center server dikelola oleh administrator. Pengelolaan data center yang baik mendukung seluruh kinerja dari jaringan, dari pemakaian aplikasi, oleh karena itu aturan dan standar pengukuran merupakan hal yang penting dari administrasi Data Center. Beberapa tahun ini data center server menjadi pembahasan yang ramai, yang sebelumnya data center server bukan merupakan bahasan yang perlu di telaah lebih dalam secara teori, tetapi kebutuhan akan informasi dari pengolahan data center yang baik membuat pakar-pakar jaringan akhirnya memutuskan untuk membahas data center server lebih dalam beserta perancangan data center dalam Infrastruktur TI sendiri.

C. Basis Data

Basis Data (Database) adalah merupakan kumpulan informasi, biasanya dalam ukuran tertentu. Contoh Basis Data yang umum adalah buku telepon. Buku ini berisi sekumpulan nama, alamat dan nomor telepon. Akan tetapi database yang akan digunakan adalah Basis Data yang disajikan dalam bentuk tabel-tabel yang terdiri atas field dan record. Field adalah bagian terkecil dari record, biasanya disediakan dalam bentuk kolom, sedangkan record adalah kumpulan field-field dalam bentuk baris. Basis data (database) merupakan sekumpulan field data yang saling berhubungan (Jeffrey.L.Whitten, 2007:518). Lingkungan basis data menekankan pada data yang tidak tergantung (independent) pada aplikasi yang menggunakan data tersebut. Basis data pada implementasi model terdiri atas tabel-tabel maka desain model data yang baik adalah memenuhi kriteria sebagai berikut :

- a. Model data yang baik harus sederhana (simple). Sesuai dengan aturan yang berlaku, atribut data terdiri atas entitas yang hanya menggambarkan entitas tersebut. Entitas adalah segala sesuatu baik nyata maupun abstrak yang ingin disimpan datanya. Sedangkan atribut data adalah karakteristik umum dari bagian tertentu entitas.
- b. Model data yang baik tidak berulang (tidak ada duplikasi data) ini berarti tidak ada atribut data selain kunci, yang menggambarkan lebih dari satu entitas.

- c. Model data yang baik harus fleksibel dan dapat beradaptasi untuk kebutuhan mendatang. Model data dirancang seindependent mungkin terhadap aplikasi yang menggunakannya.

File di dalam pemrosesan aplikasi dapat dikategorikan ke dalam beberapa tipe, diantaranya sebagai berikut :

- a. File Induk (master file)

Di dalam aplikasi, file ini merupakan file yang penting, file ini terus ada selama hidup dari sistem informasi.

- b. File Transaksi

File transaksi biasa juga disebut dengan file input (input file). File ini digunakan untuk merekam data hasil dari suatu transaksi yang terjadi.

- c. File Laporan

File ini disebut juga dengan file output. File yang berisi informasi yang akan ditampilkan. File ini dibuat untuk mempersiapkan pembuatan suatu laporan dan biasanya dilakukan bila printer belum siap atau masih digunakan oleh proses yang lain.

- d. File Sejarah

File sejarah biasa juga disebut file arsip yaitu file yang berisi data-data lalu yang tidak aktif lagi, tetapi perlu disimpan untuk keperluan yang akan datang.

e. File Pelindung

File pelindung merupakan salinan dari file-file yang masih aktif di database pada satu saat tertentu. File ini digunakan sebagai file cadangan atau pelindung apabila file database yang aktif rusak atau hilang.

f. File Kerja

File kerja disebut juga file sementara (temporary file) atau scarth file. File ini dibuat oleh suatu proses program secara sementara karena memori selama proses dan akan dihapus apabila proses telah selesai.

Antara satu tabel dengan tabel yang lain dapat saling direlasikan dengan beberapa tipe relasi yaitu :

a. Relasi satu ke satu (one to one)

Relasi satu ke satu adalah relasi dimana satu entity akan direlasikan dengan satu entity pula.

b. Relasi satu ke banyak (one to many)

Relasi satu ke banyak adalah relasi dimana satu entity direlasikan dengan banyak entity.

c. Relasi banyak ke banyak (many to many)

Relasi banyak ke banyak adalah relasi yang banyak entity direlasikan dengan banyak entity pula.

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan

dengan nilai (angka,deretan karakter, atau symbol). Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

- a. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi kebutuhan.
- c. Kumpulan file/table/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

1. Tujuan Basis Data

Basisdata bertujuan untuk mengatur data sehingga diperoleh kemudahan, ketepatan, dan kecepatan dalam pengambilan kembali. Untuk mencapai tujuan, syarat sebuah basis data yang baik adalah sebagai berikut;

a. Tujuan adanya redundansi dan inkonsistensi data

Redundansi terjadi jika suatu informasi disimpan di beberapa tempat. Misalnya, ada data mahasiswa yang memuat NIM, nama, alamat, dan atribut lainnya, sementara kita punya data lain tentang data KHS mahasiswa yang isinya yang terdapat NIM, nama, mata kuliah, dan nilai.

b. Kesulitan Pengaksesan Data

Basis data memiliki fasilitas untuk melakukan pencarian informasi dengan menggunakan Query taupun dari tool untuk melihat

tabelnya. Dengan fasilitas ini, kita bisa secara langsung melihat data dari software DBMS-nya. Selain itu, basis data bisa dihubungkan dengan program aplikasi sehingga memudahkan pengguna dalam mengakses informasi. Misalnya program aplikasi untuk kasir yang terhubung dengan basis data, pengguna cukup menggunakan fasilitas pencarian ataupun laporan. Yang tersedia pada program aplikasi untuk mendapatkan informasi stok, laporan penjualan, dan lain-lain. Dalam basis data, informasi yang diperoleh dari kumpulan data bisa berupa keseluruhan data, sebagai data, data dengan filter tertentu, data yang terurut, ataupun data summary. Sebagai contoh sederhana ketika kita ingin mencatat data alamat dan telepon dari kolega kita. Sebagai orang akan menggunakan buku alamat. Metode pencatatan dilakukan dengan menuliskan data setelah catatan terakhir. Ketika kita menginginkan informasi alamat seseorang kita akan mencari karena informasi yang tersaji tidak terurut. Ada juga orang mencatat dengan mengelompokkan nama berdasarkan abjad. Hal ini akan lebih mempermudah pencarian karena kita tidak perlu membaca keseluruhan data, tetapi cukup dalam satu kelompok saja. Tapi masalah baru muncul ketika jumlah data untuk sekelompok data abjad tertentu terlalu banyak sedangkan kelompok abjad yang lain masih terlalu sedikit. Dalam metode ini, ada banyak ruang tidak terpakai jika memberikan ruang yang sama untuk setiap kelompok. Dalam hal pencarian, kesulitan akan kita temui ketika informasi yang kita ingin cari dengan kata kunci sebagian namanya. Misalnya kita akan mencari

alamat Anto, sementara yang tercatat dalam buku catatan adalah Mardianto. Tentu saja kita tidak akan dapat menemukannya dalam kelompok data dengan huruf depan A. selain itu, tidak selamanya kata kunci yang diketahui adalah dari nama, tetapi bisa saja yang diketahui adalah nomer teleponnya, sedangkan yang ingin kita cari adalah alamat dan namanya. Hal ini merupakan masalah baru dari pencatatan data dengan buku. Basis data bisa memberikan solusi terhadap permasalahan-permasalahan tersebut diatas.

c. Multiple User

Basis data memungkinkan pengguna data bersama-sama oleh banyak pengguna pada saat yang bersamaan atau pada saat yang berbeda. Dengan meletakkan basis data pada bagian server yang bisa diakses kesemua pengguna dari banyak klient, kita sudah menyediakan akses kesemua pengguna dari computer klient ke sumber informasi yaitu basis data. Tentu saja pengaksesan oleh pengguna-pengguna ini disesuaikan dengan hak aksesnya. Misalnya sebuah perguruan tinggi memiliki data tentang mahasiswa, pembayaran, dan lain-lain yang diletakan dalam sebuah basis data. Bagian Akademi akan bisa mengakses data-data akademi mahasiswa, Bagian Keuangan akan diijinkan mengakses data pembayaran mahasiswa, sementara mahasiswa hanya bisa melihat status akademi/keuangan yang berhubungan dengan dirinya saja. Hal ini sangat dimungkinkan dengan penyimpanan data dalam basisdata.

2. Manfaat / Kelebihan Basis Data

Banyak memanfaatkan yang dapat kita peroleh dengan menggunakan basis data. Manfaat/kelebihan basis data diantaranya adalah :

a. Kecepatan dan kemudahan (speed)

Dengan menggunakan basis data pengambilan informasi dapat dilakukan dengan cepat dan mudah. Basis data memiliki kemampuan dalam mengelompokkan, mengurutkan bahkan perhitungan dengan matematika. Dengan perancangan yang benar, maka penyajian informasi akan dapat dilakukan dengan cepat dan mudah.

b. Kebersamaan Pemakai

Sebuah basis data dapat digunakan oleh banyak user dan banyak aplikasi. Untuk data-data yang diperlukan oleh banyak orang/bagian. Tidak perlu dilakukan pencatatan dimasing-masing bagian, tetapi cukup dengan satu basis data untuk dipakai bersama. Misalnya data mahasiswa dalam suatu perguruan tinggi, dibutuhkan oleh banyak bagian, diantaranya: bagian akademik, bagian keuangan, bagian kemahasiswaan, dan perpustakaan. Tidak harus semua bagian ini memiliki catatan dan semua bagian bisa mengakses data tersebut sesuai dengan keperluannya.

c. Pemusatan control data

Karena cukup dengan satu basis data untuk banyak keperluan, pengontrolan terhadap data juga cukup dilakukan di satu tempat saja. Jika ada perubahan data alamat mahasiswa misalnya, maka tidak perlu kita

meng-update semua data dimasing-masing bagian tetapi cukup hanya disatu basis data.

d. Efisiensi ruang penyimpanan (space)

Dengan pemakain bersama, kita tidak perlu menyediakan tempat penyimpanan diberbagai tempat, tetapi cukup satu saja sehingga ini akan menghemat ruang penyimpanan data yang dimiliki oleh sebuah organisasi. Dengan teknik perancangan basis data yang benar, kita akan menyederhanakan penyimpanan sehingga tidak semua data harus disimpan.

e. Keakuratan (Accuracy)

Penerapan secara ketat aturan tipe data, domain data, keunikan data, hubungan antara data, dan lain-lain, dapat menekan keakuratan dalam pemasukan/penyimpanan data.

f. Ketersediaan (availability)

Dengan basis data kita dapat mem-backup data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu kita simpan ke tempat lain. Hal ini mengingat pertumbuhan transaksi suatu organisasi dari waktu ke waktu membutuhkan media penyimpanan yang semakin besar.

g. Keamanan (Security)

Kebanyakan DBMS dilengkapi dengan fasilitas manajemen pengguna diberikan hak akses yang berbeda-beda sesuai dengan

pengguna dan posisinya. Basis data bisa diberikan passwordnya untuk membatasi orang yang mengaksesnya.

h. Kemudahan dalam pembuatan program aplikasi baru

Pengguna basis data merupakan bagian dari perkembangan teknologi. Dengan adanya basis data pembuatan aplikasi bisa memanfaatkan kemampuan dari DBMS, sehingga pembuatan aplikasi tidak perlu mengurus penyimpanan data, tetapi cukup mengatur interface untuk pengguna.

i. Pemakain secara langsung

Basis data memiliki fasilitas untuk melihat datanya secara langsung dengan tool yang disediakan oleh DBMS. Untuk melihat data, langsung ke table ataupun menggunakan query. Biasanya yang menggunakan fasilitas ini adalah user yang sudah ahli, atau database administrator.

j. Kebebasan data (Data Independence)

Jika sebuah program telah selesai dibuat, dan ternyata ada perubahan isi/struktur data. Maka dengan basis data, perubahan ini hanya perlu dilakukan pada level DBMS tanpa harus membongkar kembali program aplikasinya.

k. User view

Basis data penyediaan pandangan yang berbeda-beda untuk tiap-tiap pengguna. Misalnya kita memiliki data-data dari perusahaan yang bergerak dibidang retail. Data yang ada berupa data barang, penjualan,

dan pembelian. Ada beberapa jenis pengguna yang memerlukan informasi terkait dengan data perusahaan tersebut. Mereka adalah pelanggan, kasir, bagian gudang, bagian akuntansi dan manajer. Tidak semua data boleh diakses oleh semua pengguna. Misalnya kasir dia hanya boleh berhak melihat informasi nama barang dan harga jualnya. Sementara itu dia berhak untuk memasukan data penjualan .berbeda dengan pelanggan yang hanya melihat data keberadaan barang dan harga jual tetapi tidak berhak memasukan atau merubah data. Sementara itu bagian akuntansi berhak melihat keuntungan dari tiap-tiap barang untuk menganalisa data akutansinya. Basis data mampu memberikan layanan organisasi seperti ini.

D. Basis Data Terdistribusi

Basis data terdistribusi adalah kumpulan dari beberapa basis data yang saling berkaitan secara logik dan didistribusikan melalui jaringan computer (M.Thamer Ozsu, 2011:3). Properti yang terutama terdapat pada basis data terdistribusi :

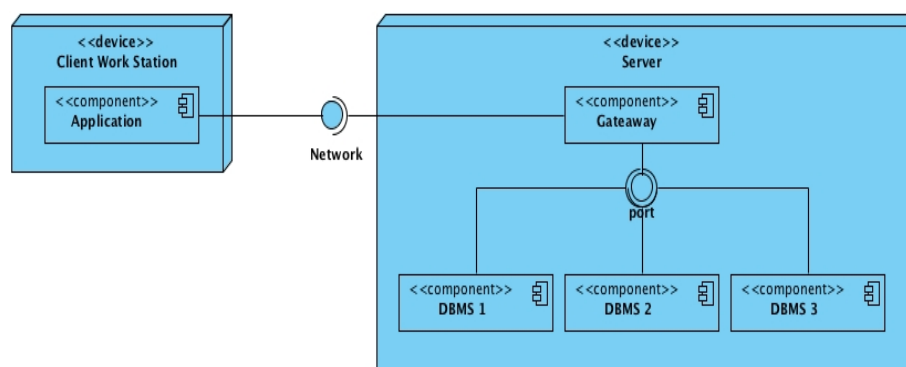
- a. Independensi data terdistribusi : pemakai tidak perlu mengetahui dimana data berada (merupakan pengembangan prinsip independensi data fisik dan logika).
- b. Transaksi terdistribusi yang atomic : pemakai dapat menulis transaksi yang mengakses dan mengubah data pada beberapa tempat seperti mengakses transaksi lokal.

Untuk trend basis data terdistribusi saat ini, pemakai harus mengetahui dimana data ditempatkan, juga harus mengetahui dimana system yang tidak mendukung independensi data terdistribusi dan transaksi terdistribusi atomic. Kedua property tersebut harus mendukung system secara efisien. Untuk system terdistribusi yang bersifat global, properti-properti tersebut kemungkinan tidak tepat karena adanya administrasi yang terlalu berlebihan dalam membuat lokasi data yang transparan.

1. Tipe Basis Data Terdistribusi

Terdapat dua tipe basis data Terdistribusi :

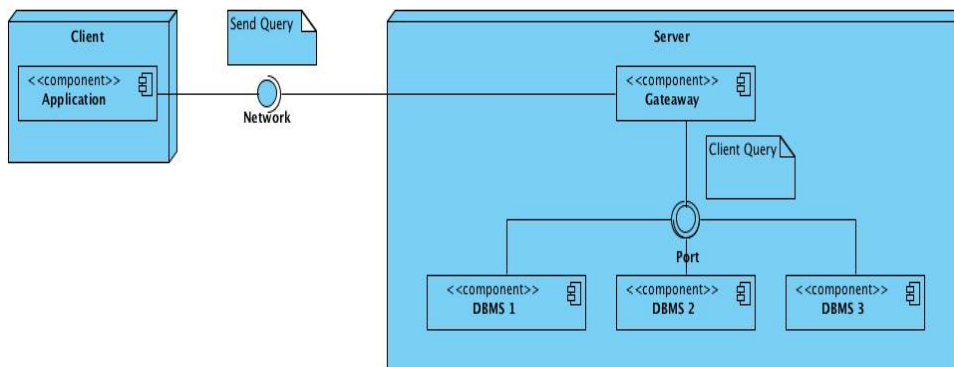
- a. Homogen : yaitu sistem dimana setiap tempat menjalankan tipe DBMS yang sama
- b. Heterogen : yaitu sistem dimana setiap tempat yang berbeda menjalankan DBMS yang berbeda, baik Relational DBMS (RDBMS) atau non relational DBMS. Gambaran basis data terdistribusi yang heterogen dapat dilihat pada Gambar di bawah ini:



Gambar 2.2: Basis Data Heterogen

2. Arsitektur Basis Data Terdistribusi

Terdapat tiga pendekatan alternatif untuk membagi fungsi pada proses DBMS yang berbeda. Dua arsitektur alternatif DBMS terdistribusi adalah Client/Server dan Collaboration Server. Arsitektur ini menjadi sangat populer untuk beberapa alasan. Pertama, implementasi yang relatif sederhana karena pembagian fungsi yang baik dan karena server tersentralisasi. Kedua, mesin server yang mahal utilitasnya tidak terpengaruh pada interaksi pemakai, meskipun mesin client tidak mahal. Ketiga, Client-Server, Sistem client-server mempunyai satu atau lebih proses client dan satu atau lebih proses server, dan sebuah proses client dapat mengirim query ke sembarang proses server seperti berikut.



Gambar 2.3: Sistem Client Server

Client bertanggung jawab pada antar muka untuk user, sedangkan server mengatur data dan mengeksekusi transaksi. Sehingga suatu proses client berjalan pada sebuah personal komputer dan mengirim query ke sebuah server yang berjalan pada mainframe. Pemakai dapat menjalankan antarmuka berbasis grafis sehingga pemakai lebih mudah dibandingkan antar muka pada server yang tidak user-friendly. Pada saat menulis aplikasi client-server, perlu diingat batasan antara client dan server dan untuk menjaga komunikasi antara keduanya yang berorientasi himpunan. Khususnya membuka kursor dan mengambil tupel pada satu waktu membangkitkan beberapa pesan dan dapat diabaikan.

Collaboration Server Arsitektur client-server tidak mengizinkan satu query mengakses banyak server karena proses client harus dapat membagi sebuah query ke dalam beberapa subquery untuk dieksekusi pada tempat yang berbeda dan kemudian membagi jawaban ke subquery. Proses client cukup kompleks dan terjadi overlap dengan server; sehingga perbedaan antara client dan server menjadi jelas. Untuk mengurangi perbedaan digunakan alternatif arsitektur client – server yaitu sistem Collaboration Server. Pada sistem ini terdapat sekumpulan server basis data, yang menjalankan transaksi data lokal yang bekerjasama mengeksekusi transaksi pada beberapa server seperti pada Gambar 7-3.. Jika server menerima query yang membutuhkan akses ke data pada server lain, sistem membangkitkan subquery yang dieksekusi server lain

dan mengambil hasilnya bersama-sama untuk menggabungkan jawaban menjadi query asal.

3. Penyimpanan data pada sistem terdistribusi

Pada DBMS terdistribusi, relasi disimpan pada beberapa tempat. Pengaksesan relasi yang disimpan pada remote side mengakibatkan biaya melewati pesan dan untuk menguranginya, sebuah relasi dipartisi atau difragmentasi ke beberapa tempat, dengan fragmen dikirim pada tempat dimana fragmen tersebut sering diakses, atau replika pada setiap tempat dimana relasi menjadi kebutuhan yang tinggi.

E. Web Service

Web service adalah suatu sistem perangkat lunak yang didesain untuk mendukung interaksi mesin ke mesin pada suatu jaringan (W3C, 2004). Web service dapat dibangun dengan menggunakan bahasa pemrograman apa saja dan juga dapat diimplementasikan pada platform manapun. Hal ini dimungkinkan karena web service berkomunikasi menggunakan sebuah standar format data yang universal yaitu XML dan menggunakan protokol SOAP. Karena web service menggunakan format data XML, maka web service juga mewariskan sifat multi-tier dari XML sehingga memungkinkan terjadinya integrasi antar web service atau aplikasi.

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam

Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

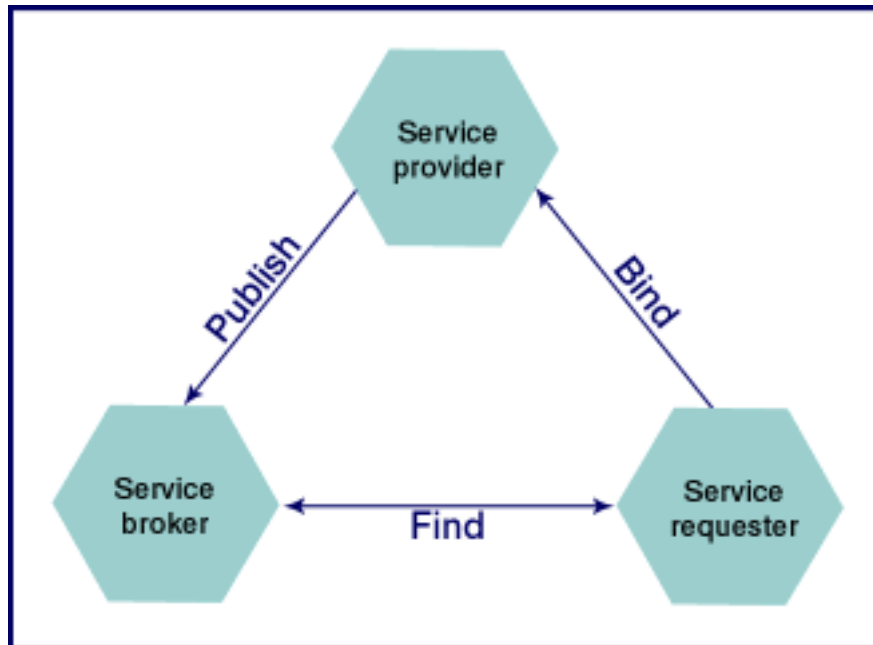
- a. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
- b. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
- c. Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

1. Arsitektur Web Service

Web service memiliki tiga entitas dalam arsitekturnya, yaitu:

- a. Service Requester (peminta layanan) :Peminta layanan yang mencari dan menemukan layanan yang dibutuhkan serta menggunakan layanan tersebut.
- b. Service Provider (penyedia layanan) :Berfungsi untuk menyediakan layanan/service dan mengolah sebuah registry agar layanan-layanan tersebut dapat tersedia.

- c. Service Registry (daftar layanan) :Berfungsi sebagai lokasi central yang mendeskripsikan semua layanan/service yang telah di-register.



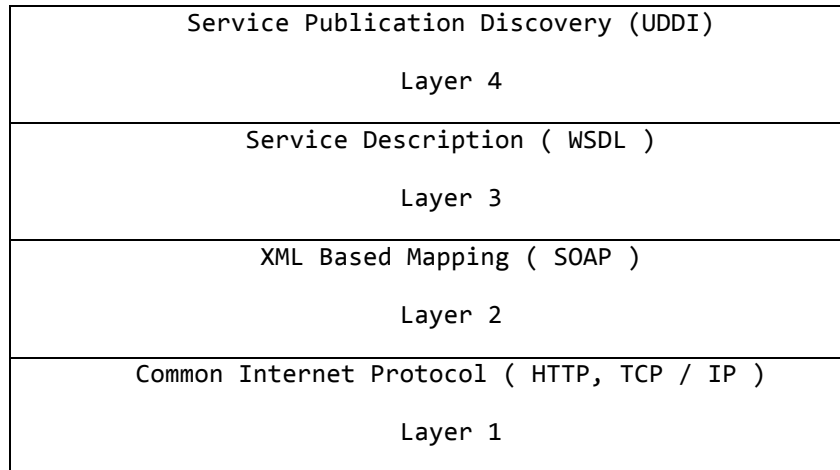
Gambar 2.4: Arsitektur Web Service (sumber ibm)

2. Operasi – Operasi Web Service

Secara umum, web service memiliki tiga operasi yang terlibat di dalamnya, yaitu:

- Publish/Unpublish: Menerbitkan/menghapus layanan ke dalam atau dari registry.
- Find: Service requestor mencari dan menemukan layanan yang dibutuhkan.
- Bind: Service requestor setelah menemukan layanan yang dicarinya, kemudian melakukan binding ke service provider untuk melakukan interaksi dan mengakses layanan/service yang disediakan oleh service provider.

3. Komponen – Komponen Web Service



Gambar 2.5: Komponen Web Service (sumber ibm)

Web service secara keseluruhan memiliki empat layer komponen seperti pada gambar di atas, yaitu:

- a. Layer 1: Protokol internet standar seperti HTTP, TCP/IP
- b. Layer 2: Simple Object Access Protocol (SOAP), merupakan protokol akses objek berbasis XML yang digunakan untuk proses pertukaran data/informasi antar layanan.
- c. Layer 3: Web Service Definition Language (WSDL), merupakan suatu standar bahasa dalam format XML yang berfungsi untuk mendeskripsikan seluruh layanan yang tersedia.
- d. Layer 4: Universal Description, Discovery and Integration (UDDI) merupakan suatu standar dalam format XML yang berfungsi sebagai deskripsi, menampilkan, dan mencari sebuah layanan web service

F. XML (Extensible Markup Language)

XML merupakan sebuah Markup Language untuk membuat dokumentasi terstruktur (Jennifer Kyrnin, 2013). Dokumen-dokumen terstruktur adalah dokumen-dokumen yang mempunyai isi/content (kata, gambar) serta indikasi yang menyatakan makna dari content tersebut. XML mempunyai kelebihan sebagai berikut:

- a. XML tidak tergantung pada platform atau sistem operasi yang digunakan.
- b. Hasil pencarian data lebih akurat.
- c. Dokumen XML dapat diterjemahkan ke dalam beberapa format yang berbeda karena dalam XML data dan instruksi dipisahkan.

G. Simple Object Access Protocol (SOAP)

SOAP (Simple Object Access Protocol) merupakan protokol yang digunakan untuk mempertukarkan data atau informasi dalam format XML (W3C, 2013). SOAP dapat dikatakan sebagai gabungan antara HTTP dengan XML karena SOAP umumnya menggunakan protokol HTTP sebagai sarana transport datanya dan data yang akan dipertukarkan ditulis dalam format XML. Karena SOAP menggunakan HTTP dan XML maka SOAP memungkinkan pihak-pihak yang mempunyai platform, sistem operasi dan perangkat lunak yang berbeda dapat saling mempertukarkan datanya.

SOAP (Simple Object Access Protocol) dapat menjadi layer tersendiri dalam sebuah protokol web service dalam hal menyediakan dasar atau acuan dalam pengiriman pesan pada web service dimana aplikasi berbasis web tersebut berada. Protocol SOAP yang dibangun berdasarkan konsep XML, terbagi atas tiga bagian:

- a. Envelope, yang menjelaskan apa yang ada dalam message yang dikirimkan dan bagaimana memprosesnya.
- b. Encoding Rule, yang digunakan untuk menyatakan datatype yang digunakan dalam memproses message tersebut di level aplikasi.
- c. Convention, yang merepresentasikan prosedur dan respon yang digunakan.

Arsitektur SOAP (*Simple Object Access Protocol*) terdiri dari beberapa layer spesifikasi yaitu untuk *format message* menggunakan *Message Exchange Patterns* (MEP), layer khusus untuk protocol transport, model message processing dan protokol tambahan. SOAP sendiri adalah pengganti dari XML-RPC (Remote Procedure Calling), namun dengan karakteristik sendiri, tiga karakteristik utama dari SOAP antara lain:

- a. Extensibility, dapat menggunakan beberapa extension tambahan lain diantaranya antara lain security dan WebService-Routing (masih dalam tahap pengembangan).
- b. Neutrality, SOAP dapat digunakan di berbagai protokol transport seperti HTTP, SMTP atau TCP.

- c. Independence, SOAP tidak memiliki ketergantungan terhadap bahasa pemrograman tertentu, sehingga SOAP dapat digunakan di berbagai model pemrograman.

SOAP pertama kali didesain oleh Dave Winer, Don Box, Bob Atkinso dan Mohsen Al-Ghosein pada tahun 1998 dalam sebuah proyek Microsoft yang bekerjasama dengan Develop Mentor dan User and Software. SOAP awalnya dibangun sebagai sebuah object access protocol dan awalnya lebih menyerupai *Internet Inter-ORB Protocol (IIOP)*, sebuah protocol yang merupakan bagian dari *Common Object Request Broker Architecture (CORBA)*.

Setelah SOAP diperkenalkan pertamakali secara umum, SOAP telah menjelma menjadi layer pokok yang lebih kompleks dari sebuah web service, berdasarkan *Web Service Description Language (WSDL)* dan *Universal Description Discovery and Integration (UDDI)*. Saat ini spesifikasi SOAP berada dibawah naungan XML Protocol Working Group dari W3C (World Wide Web Consortium).

Spesifikasi SOAP mengatur framework dari proses messaging yang terdiri dari:

- a. Processing Model, yang menjelaskan aturan dalam memproses message SOAP.
- b. Extensibility Model, menjelaskan konsep fitur SOAP serta modul SOAP itu sendiri.

- c. Underlying Protocol Binding, menjelaskan aturan yang digunakan untuk mengikat message dengan protokol poko lainnya agar dapat digunakan dalam pertukaran message antara node SOAP.
- d. Message Construct, mengatur struktur dari sebuah message SOAP. Secara struktural, SOAP terdiri dari 3 bagian yaitu: Envelope, Header dan Body

SOAP diinterpretasikan menggunakan XML. XML terpilih sebagai standar dalam format message karena XML lebih banyak digunakan oleh perusahaan besar dan banyak digunakan pada pengembangan aplikasi-aplikasi berbasis open source. Selain itu XML memiliki banyak tool tambahan yang memudahkan proses implmentasi berbasis SOAP. Namun penggunaan XML sebagai format dasar juga membawa kerugian selain keuntungan.Format XML lebih mudah dibaca oleh mata manusia, terdapat fasilitas error detection, dan tidak memiliki masalah interoperabilitas antar OS, namun memiliki kekurangan yaitu kecepatan pemrosesan yang lebih lambat dan rumit.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="DemoWebServiceService" targetNamespace="http://demo/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://demo/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<types>
<xsd:schema>
<xsd:import namespace="http://demo/"
schemaLocation="DemoWebServiceService_schema1.xsd" />
```

```

</xsd:schema>
</types>
<message name="hello">
<part element="tns:hello" name="parameters" />
</message>
<message name="helloResponse">
<part element="tns:helloResponse" name="parameters" />
</message>
<portType name="DemoWebServiceDelegate">
<operation name="hello">
<input message="tns:hello" />
<output message="tns:helloResponse" />
</operation>
</portType>
<binding name="DemoWebServicePortBinding"
type="tns:DemoWebServiceDelegate">
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
<operation name="hello">
<soap:operation soapAction="" />
<input>
<soap:body use="literal" />
</input>
<output>
<soap:body use="literal" />
</output>
</operation>
</binding>
<service name="DemoWebServiceService">
<port binding="tns:DemoWebServicePortBinding" name="DemoWebServicePort">

```

```
<soap:address
location=

"http://localhost:9081/HelloWorldWSPProject/DemoWebServiceService" />
</port>
</service>
</definitions>
```

Gambar 2.6: Contoh Format Message SOAP (sumber ibm)

Kelebihan dari protokol SOAP, antara lain:

- a. SOAP cukup fleksibel dalam hal penggunaan protokol transport yang berbeda. Memiliki standar dasar yaitu HTTP, SOAP juga mampu bekerja dalam protokol SMTP dan JMS (Java Message Service).
- b. SOAP dapat bekerja baik dalam tunnel HTTP dalam mendapatkan atau mengirimkan message, oleh karena itu SOAP juga dapat bekerja dengan baik dalam berbagai kondisi firewall dan proxy tanpa perlu melakukan modifikasi dalam protokol SOAP dan tanpa perlu modifikasi infrastruktur yang ada.

Sedangkan kekurangan dari protokol SOAP, yaitu:

- a. Karena SOAP dibangun berdasarkan konsep XML, SOAP memiliki daya pemrosesan yang lebih rendah dibandingkan dengan teknologi middleware seperti CORBA. Hal ini mungkin tidak menjadi masalah jika SOAP digunakan dalam skala kecil maupun dalam skala besar, hal ini perlu menjadi pertimbangan tersendiri.

- b. Jika menggunakan HTTP sebagai transport protocol dan tidak menggunakan WS-Addressing atau ESB (Enterprise Service Bus), klien yang terlibat dalam komunikasi bersifat fixed, hanya satu yang dapat menggunakan service. Dalam hal ini developer harus menggunakan polling, bukan lagi hanya notifikasi seperti pada kasus lain secara umumnya.

H. Integrasi Data

Integrasi Data adalah sekumpulan proses yang digunakan untuk mengeluarkan atau menangkap, resturkturisasi, pemindahan, dan pengangkutan atau penerbitan data dalam bentuk operasional atau analisa penyimpanan data, secara *real time* atau dalam bentuk *batch mode* (Anthony David Giordano, 2011:3).

Ada beberapa metode dan pola dari integrasi data berdasarkan tipe dari proses yang dilakukan diantaranya sebagai berikut:

- a. *Transactional Data Integration*, dimana berfokus pada bagaimana sebuah transaksi dibuat, diperbaharui, dimodifikasi, dan dihapus.
- b. *Business Intelegence Data Integration*, dimana berfokus pada kumpulan dari transaksi dan membentuk transaksi tersebut dalam bentuk struktur basis data yang memfasilitasi analisis.
- c. *Transactional* dan *Business Intelegence Data Integration*, direleksikan dalam model arsitektur

1. Enterprise Application Integration (EAI)

Enterprise Application Integration menyediakan transaksi integrasi data dari sistem yang berbeda – beda, baik dalam bentuk modifikasi atau dalam bentuk paket. EAI secara relative dapat berbentuk arsitektur sederhana dalam kondisi yang sempurna. Satu aplikasi dapat membuat sebuah transaksi, review, dan melakukan pembaharuan data untuk transaksi, dan pada akhirnya melakukan transaksi terkomitmen. Lingkungan aplikasi yang ada terdiri atas paket aplikasi *Enterprise Resource Planning* (ERP) dari penyedia seperti SAP, Oracle dan lainnya, sebagai tambahan pada aplikasi yang dimodifikasi untuk kepentingan internal.

2. Service-Oriented Architecture (SOA)

Service-Oriented Architecture (SOA) adalah pola transaksi data terintergrasi yang mengatur pesan untuk memicu sebuah objek yang akan bekerja pada level yang berbeda pada *interface* jaringan yang sama yang disebut juga *service bus*. Objek objek ini mewakili bentuk komponen – komponen fungsi bisnis yang mana dimana dibuat atau diturunkan pada beberapa lapisan *granularity*.

SOA lebih tepat dipandang sebagai beberapa framework yang memiliki beberapa komponen yang dapat saling berinteraksi dalam sebuah jaringan. SOA menyediakan sekumpulan prinsip yang menuntun pada *governing concepts* dimana konsep ini digunakan pada saat sistem sedang dikembangkan dan di integrasikan. SOA adalah sebuah kerangka

kerja yang dibungkus sebagai komponen yang berfungsi sebagai *interoperable service*: komponen – komponen yang berada diluar atau didalam *firewall* yang dapat menyediakan sebuah layanan yang dapat diintegrasikan atau digunakan oleh organisasi lain, meskipun organisasi lain tersebut menggunakan *client systems* yang berbeda.

Dari sudut pandang operasional, SOA membutuhkan beberapa layanan yang bersifat saling lepas bersama dengan sistem operasi dan pemanfaatan teknologi lain dalam sebuah kerangka kerja. kerangka kerja ini mengarahkan, mengatur dan mengorkestrai komponen dari SOA atau fungsionalitas bisnis yang dipicu oleh sebuah permintaan layanan.

3. Federation

Federation mengkombinasikan data yang berbeda kedalam bentuk logika struktur data yang sama, biasanya sebuah database yang saling berelasi, tidak memindahkan data, tetapi dengan menyediakan bentuk yang seragam dari data. Ide dasarnya adalah menghubungkan teknologi database yang beragam melalui konsep sebuah jembatan untuk menyediakan sebuah database yang bersifat virtual. Menghubungkan database pada level tabel, sehingga menyediakan kemampuan mengembangkan model data logik diseluruh organisasi tanpa memandang lokasi dan teknologi yang digunakan diseluruh jaringan.

4. Extract, Transform, Load (ETL)

ETL adalah kumpulan dan agregasi dari data yang bersifat transactional, dimana data di pilah dari beberapa sumber yang

disesuaikan kedalam sebuah basis data yang digunakan untuk pelaporan dan analisa. Biaya pembuatan dan perawatan dari proses kompleks integrasi data muncul pada ruang untuk pemindahan data dalam jumlah besar.

I. Dasar hukum PDPT

Dasar hukum pangkalan data perguruan tinggi yaitu:

- a. Keputusan Menteri Pendidikan Nasional Nomor 234/U/2000 Tentang Pedoman Pendirian Perguruan Tinggi
- b. Keputusan Menteri Pendidikan Nasional Nomor 184/U/2001 Tentang Pedoman Pengawasan – Pengendalian Dan Pembinaan Program Diploma, Sarjana Dan Pasca Sarjana Di Perguruan Tinggi.
- c. SK Ditjen Pendidikan Tinggi Nomor 34/DIKTI/Kep/2002 Tentang Perubahan Dan Peraturan Tambahan Kepmen No: 184/U/2001.
- d. UU No 20/2003 Tentang Pengawasan Pendidikan Tinggi Bersifat Horizontal.
- e. PP No 19 Tahun 2005 Tentang Standar Nasional Pendidikan.
- f. PP No 17 Tahun 2010 Tentang Pengelolaan dan Penyelenggaraan Pendidikan.
- g. SK No. 34/DIKTI/KEP/2002 Tentang DALWASBIN Program Studi.

J. Penelitian Terkait

Ada beberapa penelitian terdahulu yang meneliti tentang integrasi sistem antara lain :

1. Information Integration Using Logical Views (1997)

Penelitian ini menitik beratkan pemaparan beberapa ide mengenai penggunaan *tools* sebagai bentuk konstruksi jawaban melalui query menggunakan *views* pada *Database Management System* (Jeffrey D.Ullman, 1997).

2. Data Integration against Multiple Evolving Autonomous Schemata (2001)

Penelitian ini menitik beratkan pada integrasi data yang dihasilkan dari pendekatan multidatabase, mediasi, data warehousing, system informasi global, dan pendekatan pencocokan manajemen model / skema, dimana secara arsitektural dapat dikategorikan dalam penggabungan satu *global schema* dan yang tidak dapat digabung, sementara untuk menghadapi kendala pada *inter schema* kebanyakan dapat dikategorikan integrasi *global as a view* dan *local as a view*, Dimana kedua pendekatan mempunyai kelebihan dan kekurangan.(Christop Koch, 2001).

3. A Practical Approach To Merging Multi Dimentional Data Models

Penelitian ini menitik beratkan penggabungan dua buah atau lebih skema sehingga menghasilkan satu *global data source*, dengan menggunakan metodologi star untuk melakukan penggabungan skema, dimana contoh kasus menggunakan kasus penggabungan beberapa *data marts* menjadi sebuah konsolidasi *data warehouse* berdasarkan model manajemennya (Michael Mireku Kwaye, 2011).

Sedangkan penelitian terdahulu yang meneliti tentang teknologi web service diantaranya :

1. The Implications and Impacts of Web Services to Electronic Commerce Research and Practices

Penelitian ini menitikberatkan implementasi web service pada aplikasi e-commerce dimana penelitian ini menunjukkan e-commerce dengan menggunakan teknologi web service dapat meningkatkan pengembangan produktifitas, dan secara cepat dan mudah di integrasikan dengan partner bisnis. (Minder Chen, Andrew N.K Chen, Benjamin B.M.Shao, 2003)

2. A Survey on Web Service Composition

Penelitian ini mendiskusikan pentingnya komposisi web service, teknologi yang dibutuhkan untuk melaksanakan pelayanan web service, dan strategi komposisi berdasarkan platform dan frameworks. (Schahram Dustdar dan Wolfgang Schreiner, 2005)

3. Web Service Selection Based On Ranking of QoS Using Associative Classification

Penelitian ini menitik beratkan keseluruhan seleksi pelayanan dan ranking framework dimana menklasifikasikan kandidat web service berdasarkan level Quality Of Service berdasarkan permintaan dan kriteria pengguna dengan menggunakan Associative Classification Algorithm dan menentukan rangking berdasarkan kualitas fungsional melalui persamaan

semantic. (Molood Makhluhian, Seyyed Mohsen Hashemi, Yousef Rastegari and Emad Pejman, 2012).

Yang membedakan antara penelitian ini dengan beberapa penelitian yang disebutkan sebelumnya bahwa penelitian tidak membahas bentuk strategi komposisi dari sebuah web service, tidak juga membahas bagaimana memanfaatkan salah satu fitur dari sebuah DBMS seperti fitur views, tidak juga merupakan penggabungan beberapa skema menjadi sebuah skema global, tetapi membahas pengaruh pemanfaatan web service sebagai alat bantu proses pelaporan data dari satu institusi ke institusi lain dimana skema datanya diatur oleh sebuah skema global yang telah ditetapkan sebelumnya, skema global yang dimaksud adalah skema yang dikeluarkan oleh DITJEN-DIKTI dalam bentuk dokumen kamus data pendidikan serta menguji ketersediaan dan fungsionalitas web service yang telah dibangun oleh PDPT-DIKTI.

K. Kerangka konseptual

Direktorat Jenderal Pendidikan Tinggi telah berhasil mengembangkan sistem pelaporan data semesteran Perguruan Tinggi yang dapat mengakomodir keberagaman tingkat ketersediaan sistem informasi di masing-masing Perguruan Tinggi. Aplikasi tersebut berfungsi untuk membantu Ditjen DIKTI dalam penyediaan data dan informasi yang akurat berkaitan dengan pendidikan tinggi dalam rangka menentukan kebijakan – kebijakan serta membantu integrasi data dan informasi (Perguruan Tinggi, Kopertis, BAN-PT, dan lain lain)



Proses pelaporan data kegiatan akademik dari program studi ke Pangkalan Data Perguruan Tinggi sangat memakan waktu sehingga membuat pelaporan Pangkalan Data Perguruan Tinggi ke Direktorat Jenderal Pendidikan Tinggi menjadi terhambat



Solusi yang ditawarkan atas permasalahan tersebut adalah dibuatnya suatu aplikasi sistem informasi yang mempunyai layanan webservice dengan protocol SOAP

Applikasi Sistem Informasi ini nantinya akan membuat aliran data dari program studi ke Pangkalan Data Perguruan Tinggi, dan aliran data dari Pangkalan Data Perguruan Tinggi ke Direktorat Jenderal Pendidikan Tinggi, begitu pula sebaliknya, sehingga integrasi data dari Perguruan Tinggi ke Direktorat Jenderal Pendidikan Tinggi dapat tercapai



Diharapkan dengan adanya Applikasi Sistem Informasi yang terintegrasi ke Direktorat Jenderal Pendidikan Tinggi akan membantu dalam rangka menentukan kebijakan dan membantu integrasi data dan informasi, disisi lain bagi Perguruan Tinggi sendiri Applikasi ini dapat membantu dalam pelaksanaan Sistem Penjaminan Mutu Internal (SPMI) dan Sistem Penjaminan Mutu Eksternal (SPME)

BAB III

METODE PENELITIAN

A. Jenis dan Sumber Data

Data yang digunakan dalam penelitian ini terdiri dari data *primer* dan data *sekunder*. Data primer digunakan sebagai data *warehouse* pada sistem yang dikembangkan sedangkan data sekunder digunakan sebagai data penunjang atau referensi dalam pengembangan sistem.

1. Data Primer

Data *primer* diperoleh dari aplikasi Evaluasi Program Studi Berbasis Evaluasi Diri (EPSBED) milik program studi teknik elektro S1 dan teknik informatika S1 universitas hasanuddin. Data ini merupakan data kuantitatif, dimana data ini digunakan sebagai data pelaporan dari program studi pada sebuah perguruan tinggi ke Dirjen DIKTI.

2. Data Sekunder

Data sekunder terdiri dari referensi kepustakaan dalam hal ini dokumen kamus data pendidikan, dokumen pemetaan epsbed dengan evaluasi, buku manual pdpt versi 1.2.2, dokumen pelatihan pdpt sesi 1, 2, dan 3, dokumen teknis webservice push pdpt versi 1.4, data tabel referensi, serta buku, jurnal, dan literature diinternet yang dianggap menunjang pengembangan sistem.

B. Analisis Data

Analisa data mengacu pada dokumen kamus data pendidikan, dari analisa tersebut bahwa kategori data dalam skema PDPT dibedakan menjadi tiga kategori yaitu :

- a. Data Referensi adalah data yang bersifat independen (jarang berubah datanya) dimana data ini diakses oleh tabel referensi, tabel master dan transaksi, penamaan tabel ini selalu diawali dengan kata **TREF_***, contoh data referensi adalah : provinsi, kota, status mahasiswa, dan lain sebagainya
- b. Data Master adalah data yang jarang berubah dan mengakses data referensi serta diakses oleh data transaksi, penamaan tabel ini diawali dengan kata **TMST_***, contoh data master adalah perguruan tinggi, mahasiswa, dosen, matakuliah dan lain sebagainya
- c. Data Transaksi adalah data yang sering berubah, penamaan tabel ini diawali dengan kata **TRAN_***, contoh data transaksi adalah aktivitas mengajar dosen, aktivitas kuliah mahasiswa, publikasi dosen tetap dan lain sebagainya.

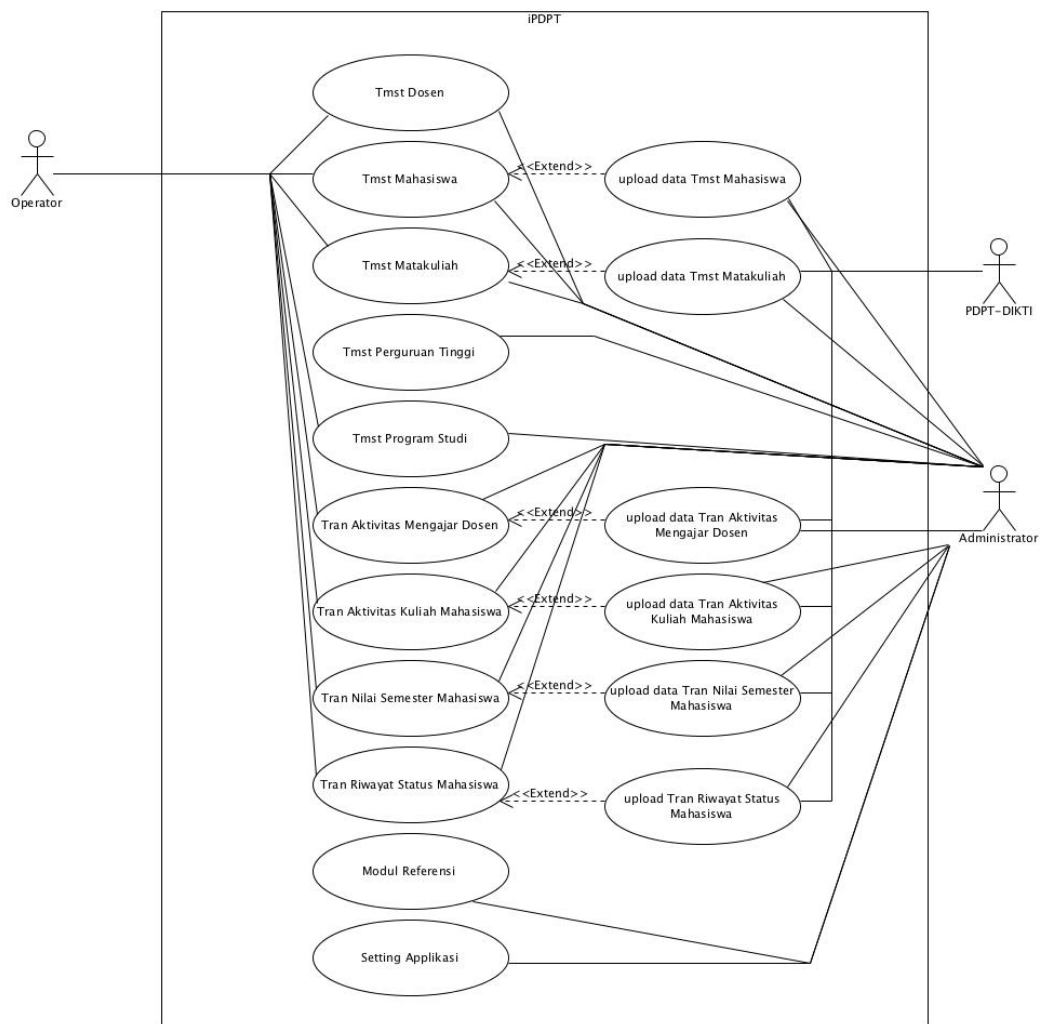
C. Perancangan Sistem

Sistem yang akan dikembangkan dalam penelitian ini berbasis web dan melibatkan pengguna dan pengelola sistem. Pengguna sistem dalam hal ini adalah operator data program studi dan pengelola sistem adalah Administrator pangkalan data perguruan tinggi. Perancangan ini menggunakan pemodelan *Unified Modeling Language (UML)* yang

merupakan metode pemodelan berbasis objek dan *Entity Relationship Diagram*, Berikut diagram perancangan sistem.

1. Use Case Diagram

Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem PDPT-DIKTI, pengguna dan administrator.



Gambar 3.1: Use Case Diagram Sistem

Pada use case ini terdapat tiga aktor utama yaitu Operator, Administrator dan sistem PDPT-DIKTI dimana use case yang dapat diakses oleh

masing - masing aktor berbeda, use case yang dapat diakses oleh operator adalah use case Tmst Dosen, Tmst Mahasiswa, Tmst Matakuliah, Tmst Program Studi, Tran Aktivitas Mengajar Dosen, Tran Aktivitas Kuliah Mahasiswa, Tran Nilai Semester Mahasiswa dan Tran Riwayat Status Mahasiswa, use case yang dapat diakses oleh PDPT-DIKTI adalah upload data Tmst Mahasiswa, upload data Tmst Matakuliah, upload data Tran Aktivitas Kuliah Mahasiswa, upload data Tran Nilai Semester Mahasiswa, upload data Tran Riwayat Status Mahasiswa, upload data Tran Aktivitas Mengajar Dosen, sedangkan administrator dapat mengakses semua use case.

2. Class Diagram

Class Diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis diantara objek tersebut. Diagram ini mendefinisikan kelas-kelas yang akan dibuat untuk membangun system.

dari sebuah modul yang menyediakan fungsi – fungsi dasar yang dibutuhkan, jika sebuah controller membutuhkan data pada database maka controller akan menghubungi class model melalui fungsi loadModel, dimana class model adalah sebuah objek yang mewakili sebuah entitas pada *database*. Pada class diagram sistem ini terdapat beberapa class controller dan class model yang dibutuhkan untuk mengirimkan data dari perguruan tinggi ke PDPT-DIKTI, class – class tersebut diantaranya adalah :

a. Class TmstPerguruanTinggiController dan Class TmstPerguruan Tinggi

Class TmstPerguruanTinggiController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul perguruan tinggi, sedangkan Class TmstPerguruanTinggi adalah class model yang menghubungkan antara Class TmstPerguruanTinggiController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TmstPerguruanTinggiController, Class TmstPerguruanTinggi mempunyai *composition* dengan Class TmstProgramStudi dan Class TmstMatakuliah.

b. Class TmstProgramStudiController dan Class TmstProgramStudi

Class TmstProgramStudiController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah

proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul program studi, sedangkan Class TmstProgramStudi adalah class model yang menghubungkan antara Class TmstProgramStudiController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TmstProgramStudiController, Class TmstProgramStudi mempunyai *composition* dengan Class TmstDosen, Class TmstMahasiswa, dan Class TmstPerguruanTinggi.

c. Class TmstMatakuliahController dan Class TmstMatakuliah

Class TmstMatakuliahController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul matakuliah, sedangkan Class TmstMatakuliah adalah class model yang menghubungkan antara Class TmstMatakuliahController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TmstMatakuliahController, Class TmstMatakuliah mempunyai *composition* dengan Class TmstPerguruanTinggi, Class TranNilaiSemesterMahasiswa, dan Class TranAktivitasMengajarDosen.

d. Class TmstDosenController dan Class TmstDosen

Class TmstDosenController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah

data, lihat detail data, ubah data dan menghapus data untuk modul dosen, sedangkan Class TmstDosen adalah class model yang menghubungkan antara Class TmstDosenController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TmstDosenController, Class TmstDosen mempunyai *composition* dengan Class TmstProgramStudi dan Class TranAktivitasMengajarDosen.

e. Class TranAktivitasMengajarDosenController dan Class TranAktivitasMengajarDosen

Class TranAktivitasMengajarDosenController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul aktivitas mengajar dosen, sedangkan Class TranAktivitasMengajarDosen adalah class model yang menghubungkan antara Class TranAktivitasMengajarDosenController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TranAktivitasMengajarDosenController, Class TranAktivitasMengajarDosen mempunyai *composition* dengan Class TmstDosen dan Class TmstMatakuliah.

f. Class TmstMahasiswaController dan Class TmstMahasiswa

Class TmstMahasiswaController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses

tambah data, lihat detail data, ubah data dan menghapus data untuk modul mahasiswa, sedangkan Class TmstMahasiswa adalah class model yang menghubungkan antara Class TmstMahasiswaController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TmstMahasiswaController, Class TmstMahasiswa mempunyai *composition* dengan Class TmstProgramStudi, Class TranRiwayatStatusMahasiswa dan Class TranAktivitasKuliahMahasiswa.

g. Class TranRiwayatStatusMahasiswaController dan Class TranRiwayatStatusMahasiswa

Class TranRiwayatStatusMahasiswaController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul riwayat status mahasiswa, sedangkan Class TranRiwayatStatusMahasiswa adalah class model yang menghubungkan antara Class TranRiwayatStatusMahasiswaController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TranRiwayatStatusMahasiswaController, Class TranRiwayatStatusMahasiswa mempunyai *composition* dengan Class TmstMahasiswa.

h. Class TranAktivitasKuliahMahasiswaController dan TranAktivitasKuliahMahasiswa

Class TranAktivitasKuliahMahasiswaController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul transaksi aktivitas kuliah mahasiswa, sedangkan Class TranAktivitasKuliahMahasiswa adalah class model yang menghubungkan antara Class TranAktivitasKuliahMahasiswaController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class TranAktivitasKuliahMahasiswaController, Class TranAktivitasKuliahMahasiswa mempunyai *composition* dengan Class TmstMahasiswa dan Class TranNilaiSemesterMahasiswa.

i. Class TranNilaiSemesterMahasiswaController dan TranNilaiSemesterMahasiswa

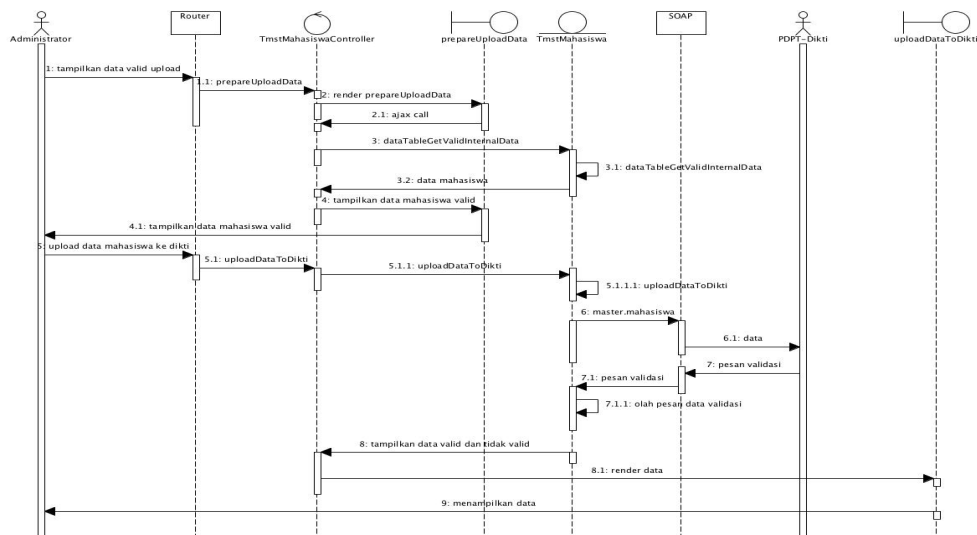
Class TranNilaiSemesterMahasiswaController adalah class yang mengatur proses logik dari sistem dimana beberapa proses logik tersebut adalah proses tambah data, lihat detail data, ubah data dan menghapus data untuk modul transaksi nilai semester mahasiswa, sedangkan Class TranNilaiSemesterMahasiswa adalah class model yang menghubungkan antara Class TranNilaiSemesterMahasiswaController dengan database baik itu untuk proses penambahan data, perubahan data, penghapusan data, melihat data dan kebutuhan proses lain yang dibutuhkan oleh Class

TranNilaiSemesterMahasiswaController, Class TranNilaiSemester Mahasiswa mempunyai *composition* dengan Class TmstMatakuliah dan Class TranAktivitasKuliahMahasiswa.

3. Sequence Diagram

Sequence diagram memodelkan interaksi objek didalam sebuah *Use Case* diagram. Diagram ini menggambarkan perilaku objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* atau *method* yang dikirimkan dan diterima antar objek. *Sequence* berikut akan menjelaskan proses pengiriman data dari system ke PDPT-DIKTI

a. Sequence Diagram pengiriman data mahasiswa

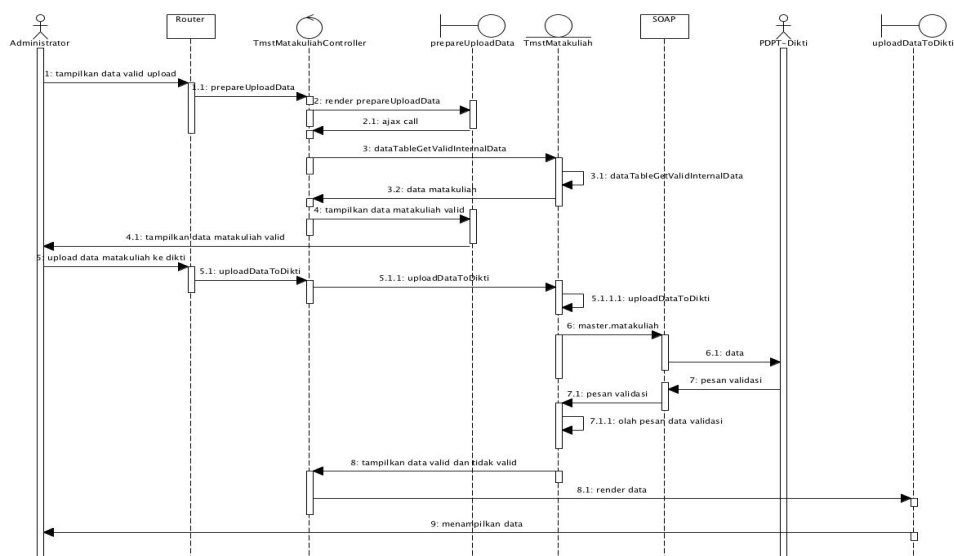


Gambar 3.3: *Sequence* Diagram pengiriman data mahasiswa

Proses pengiriman data mahasiswa dimulai dari administrator yang mengirimkan sebuah request ke sistem untuk menampilkan data

mahasiswa yang valid untuk diupload ke PDPT-DIKTI, sistem lalu menampilkan data mahasiswa yang telah memenuhi syarat validasi internal kepada administrator, setelah administrator selesai memeriksa data mahasiswa yang telah memenuhi validasi internal maka administrator mengirimkan sebuah request ke system untuk mengirimkan data tersebut ke PDPT-DIKTI, sistem lalu menggunakan fasilitas web service yang disediakan oleh PDPT-DIKTI untuk mengirimkan data, PDPT-DIKTI kemudian menerima data yang dikirimkan oleh sistem lalu memvalidasi data tersebut, setelah data divalidasi maka PDPT-DIKTI akan memberikan status validasi pada setiap data sebagai respon dari proses pengiriman data kepada sistem, sistem lalu menampilkan hasil dari respon PDPT-DIKTI kepada administrator.

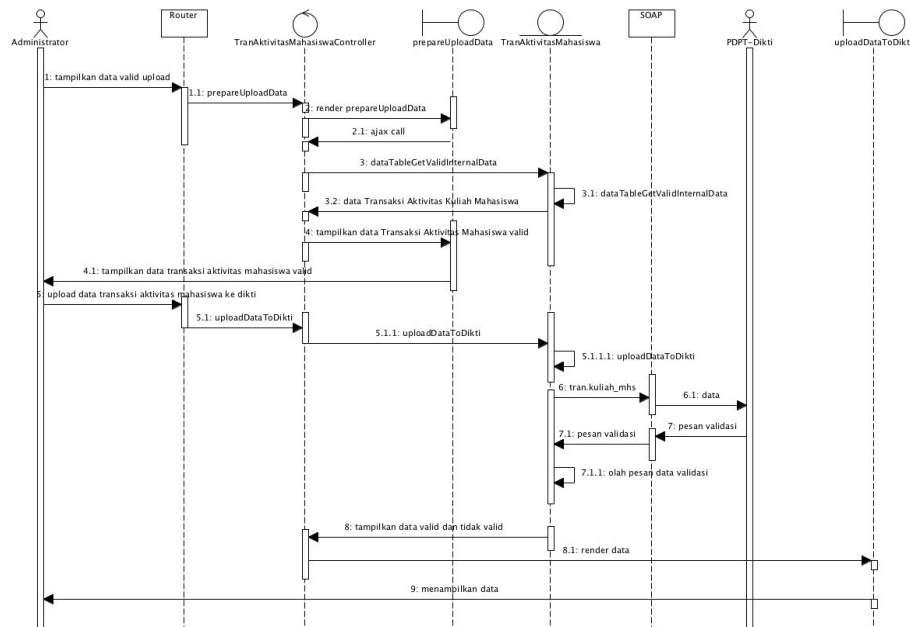
b. Sequence Diagram pengiriman data matakuliah



Gambar 3.4: Sequence Diagram pengiriman data matakuliah

Proses pengiriman data matakuliah dimulai dari administrator yang mengirimkan sebuah request ke system untuk menampilkan data matakuliah yang valid untuk diupload ke PDPT-DIKTI, system lalu menampilkan data matakuliah yang telah memenuhi syarat validasi internal kepada administrator, setelah administrator selesai memeriksa data matakuliah yang telah memenuhi validasi internal maka administrator mengirimkan sebuah request ke system untuk mengirimkan data tersebut ke PDPT-DIKTI, sistem lalu menggunakan fasilitas web service yang disediakan oleh PDPT-DIKTI untuk mengirimkan data, PDPT-DIKTI kemudian menerima data yang dikirimkan oleh sistem lalu memvalidasi data tersebut, setelah data divalidasi maka PDPT-DIKTI akan memberikan status validasi pada setiap data sebagai respon dari proses pengiriman data kepada system, system lalu menampilkan hasil dari respon PDPT-DIKTI kepada administrator.

c. Sequence Diagram pengiriman aktivitas kuliah mahasiswa

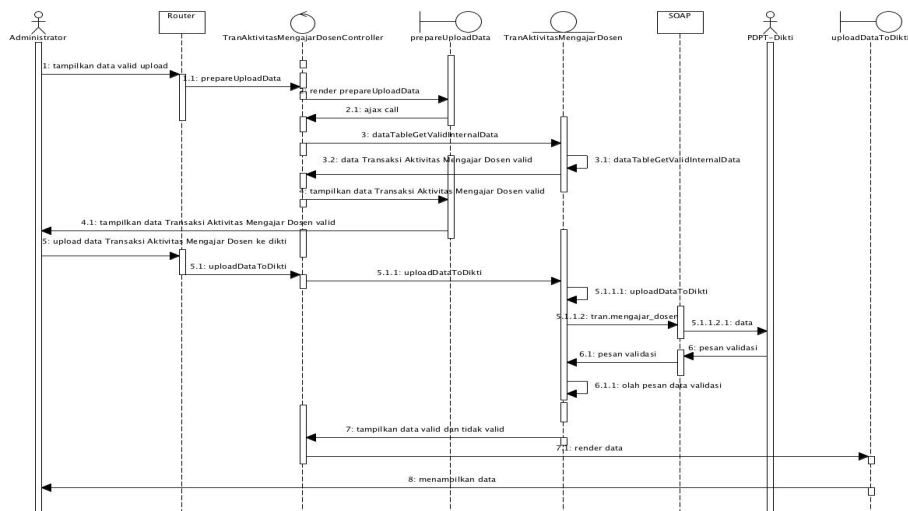


Gambar 3.5: Sequence Diagram pengiriman data mahasiswa

Proses pengiriman data transaksi aktivitas kuliah mahasiswa dimulai dari administrator yang mengirimkan sebuah request ke system untuk menampilkan data transaksi aktivitas kuliah mahasiswa yang valid untuk diupload ke PDPT-DIKTI, system lalu menampilkan data transaksi aktivitas kuliah mahasiswa yang telah memenuhi syarat validasi internal kepada administrator, setelah administrator selesai memeriksa data transaksi aktivitas kuliah mahasiswa yang telah memenuhi validasi internal maka administrator mengirimkan sebuah request ke system untuk mengirimkan data tersebut ke PDPT-DIKTI, sistem lalu menggunakan fasilitas web service yang disediakan oleh PDPT-DIKTI untuk

mengirimkan data , PDPT-DIKTI kemudian menerima data yang dikirimkan oleh system lalu memvalidasi data tersebut, setelah data divalidasi maka PDPT-DIKTI akan memberikan status validasi pada setiap data sebagai respon dari proses pengiriman data kepada sistem, sistem lalu menampilkan hasil dari respon PDPT-DIKTI kepada administrator.

d. Sequence Diagram pengiriman data mengajar dosen

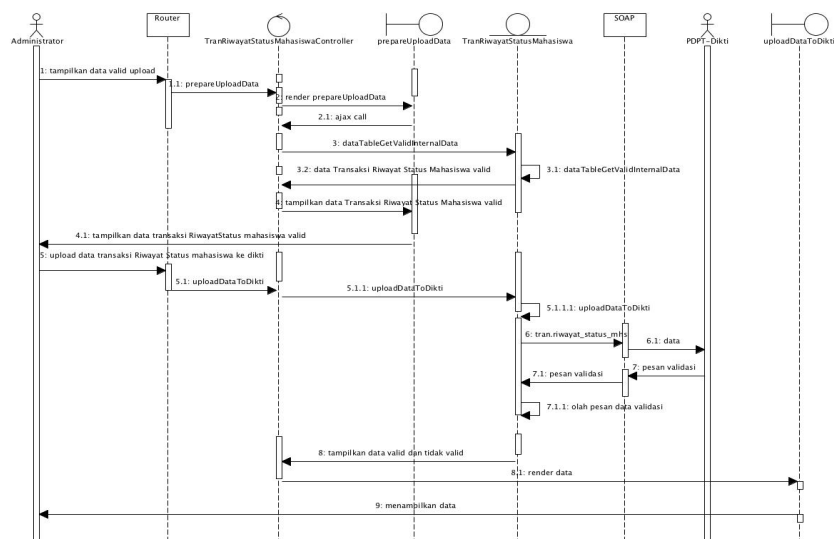


Gambar 3.5: Sequence Diagram pengiriman data mengajar dosen

Proses pengiriman data transaksi aktivitas mengajar dosen dimulai dari administrator yang mengirimkan sebuah request ke system untuk menampilkan data transaksi aktivitas mengajar dosen yang valid untuk diupload ke PDPT-DIKTI, sistem lalu menampilkan data transaksi aktivitas mengajar dosen yang telah memenuhi syarat validasi internal kepada administrator, setelah administrator selesai memeriksa data transaksi aktivitas mengajar dosen yang telah memenuhi validasi internal

maka administrator mengirimkan sebuah request ke system untuk mengirimkan data tersebut ke PDPT-DIKTI, system lalu menggunakan fasilitas web service yang disediakan oleh PDPT-DIKTI untuk mengirimkan data , PDPT-DIKTI kemudian menerima data yang dikirimkan oleh system lalu memvalidasi data tersebut, setelah data divalidasi maka PDPT-DIKTI akan memberikan status validasi pada setiap data sebagai respon dari proses pengiriman data kepada system, system lalu menampilkan hasil dari respon PDPT-DIKTI kepada administrator.

e. Sequence Diagram pengiriman data transaksi riwayat status mahasiswa

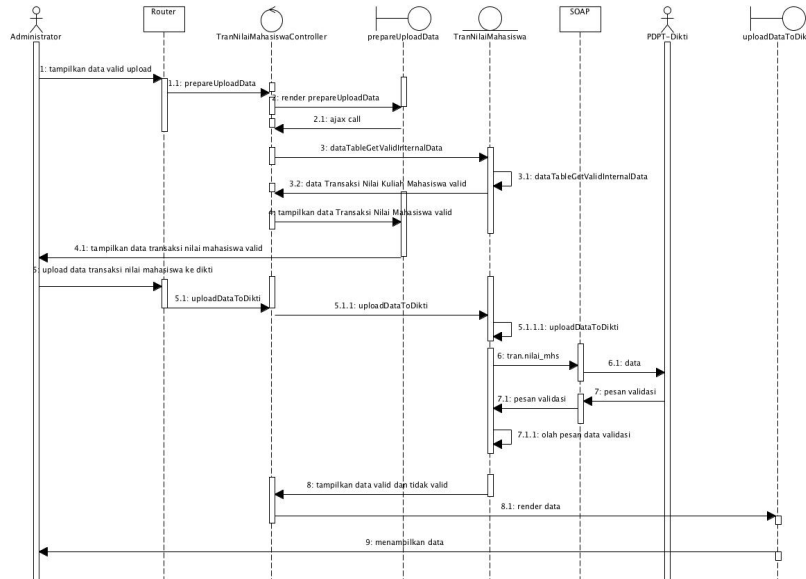


Gambar 3.6: Sequence Diagram pengiriman data transaksi riwayat status mahasiswa

Proses pengiriman data transaksi riwayat status mahasiswa dimulai dari administrator yang mengirimkan sebuah request ke system untuk menampilkan data transaksi aktivitas kuliah mahasiswa yang valid untuk

diupload ke PDPT-DIKTI, system lalu menampilkan data transaksi riwayat status mahasiswa yang telah memenuhi syarat validasi internal kepada administrator, setelah administrator selesai memeriksa data transaksi riwayat status mahasiswa yang telah memenuhi validasi internal maka administrator mengirimkan sebuah request ke system untuk mengirimkan data tersebut ke PDPT-DIKTI, sistem lalu menggunakan fasilitas web service yang disediakan oleh PDPT-DIKTI untuk mengirimkan data , PDPT-DIKTI kemudian menerima data yang dikirimkan oleh system lalu memvalidasi data tersebut, setelah data divalidasi maka PDPT-DIKTI akan memberikan status validasi pada setiap data sebagai respon dari proses pengiriman data kepada sistem, sistem lalu menampilkan hasil dari respon PDPT-DIKTI kepada administrator.

f. Sequence Diagram pengiriman data transaksi nilai mahasiswa

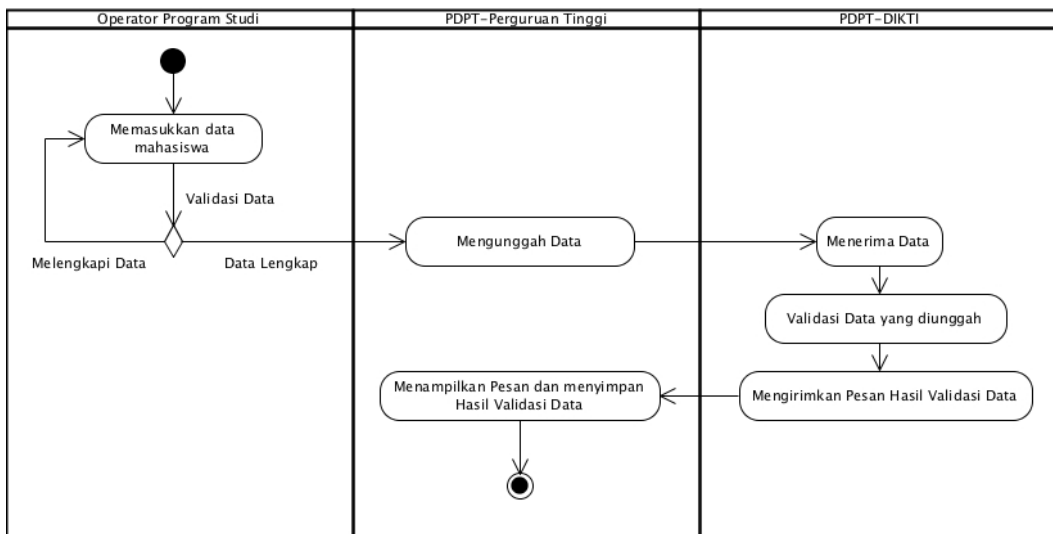


Gambar 3.7: Sequence Diagram pengiriman data transaksi nilai mahasiswa

Proses pengiriman data transaksi nilai mahasiswa dimulai dari administrator yang mengirimkan sebuah request ke system untuk menampilkan data transaksi aktivitas kuliah mahasiswa yang valid untuk diupload ke PDPT-DIKTI, sistem lalu menampilkan data transaksi nilai mahasiswa yang telah memenuhi syarat validasi internal kepada administrator, setelah administrator selesai memeriksa data nilai mahasiswa yang telah memenuhi validasi internal maka administrator mengirimkan sebuah request ke system untuk mengirimkan data tersebut ke PDPT-DIKTI, sistem lalu menggunakan fasilitas web service yang disediakan oleh PDPT-DIKTI untuk mengirimkan data , PDPT-DIKTI kemudian menerima data yang dikirimkan oleh system lalu memvalidasi

data tersebut, setelah data divalidasi maka PDPT-DIKTI akan memberikan status validasi pada setiap data sebagai respon dari proses pengiriman data kepada sistem, sistem lalu menampilkan hasil dari respon PDPT-DIKTI kepada administrator.

g. Activity diagram pengiriman data mahasiswa

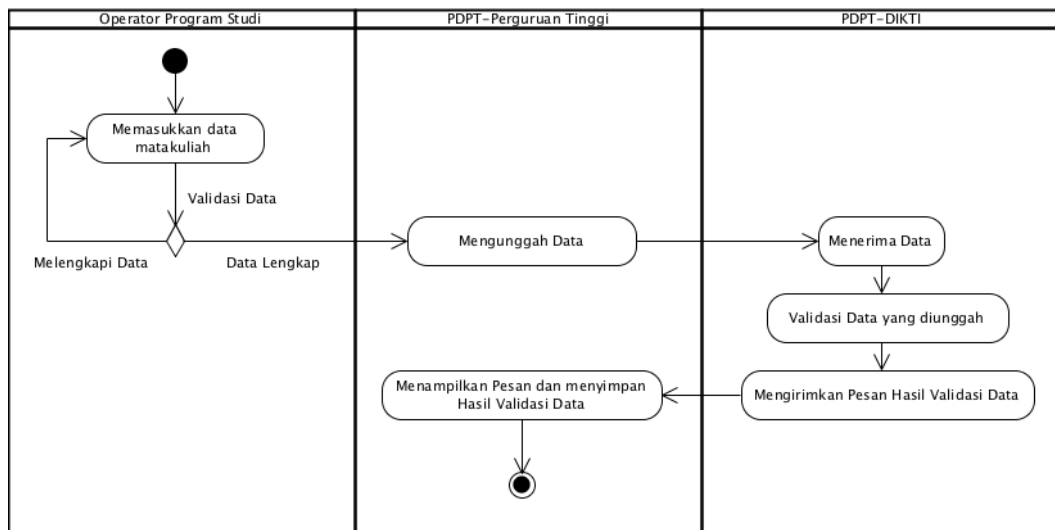


Gambar 3.8: Activity diagram pengiriman data mahasiswa

Proses pengiriman data mahasiswa dimulai dari operator program studi memasukkan data mahasiswa kedalam system, data yang telah diinput oleh operator program studi dan divalidasi oleh system kemudian dipersiapkan oleh administrator untuk diunggah ke PDPT-DIKTI, setelah administrator mengunggah data ke PDPT-DIKTI, PDPT-DIKTI lalu memvalidasi data yang dikirim oleh system, setelah divalidasi PDPT-DIKTI akan mengirimkan pesan validasi kepada system sebagai bentuk respon dari hasil pengiriman data, pesan tersebut lalu ditampilkan kepada

administrator dan disimpan kode validasinya ke atribut validasi pada data mahasiswa yang divalidasi.

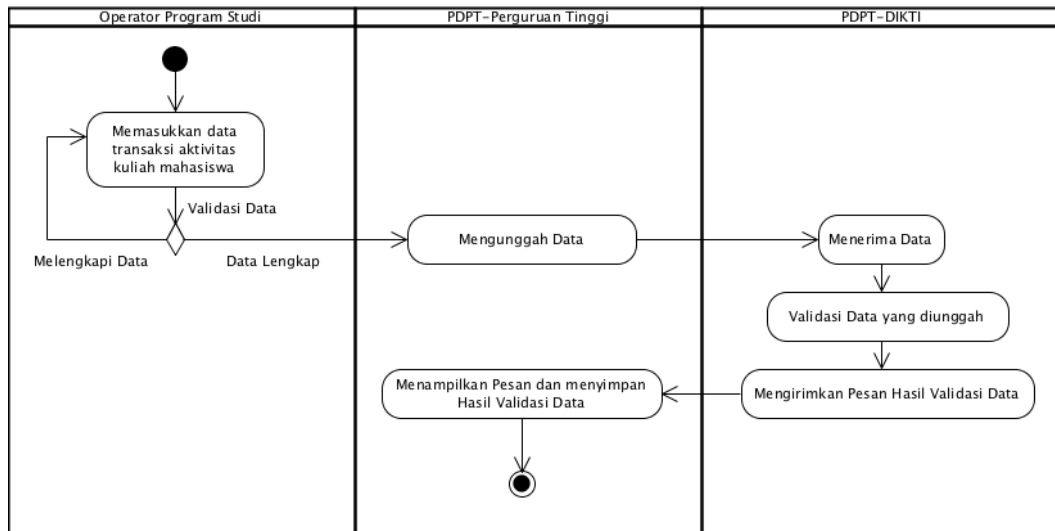
h. Activity diagram pengiriman data matakuliah



Gambar 3.9: Activity diagram pengiriman data matakuliah

Proses pengiriman data matakuliah dimulai dari operator program studi memasukkan data matakuliah kedalam system, data yang telah diinput oleh operator program studi dan divalidasi oleh system kemudian dipersiapkan oleh administrator untuk diunggah ke PDPT-DIKTI, setelah administrator mengunggah data ke PDPT-DIKTI, PDPT-DIKTI lalu memvalidasi data yang dikirim oleh system, setelah divalidasi PDPT-DIKTI akan mengirimkan pesan validasi kepada system sebagai bentuk respon dari hasil pengiriman data, pesan tersebut lalu ditampilkan kepada administrator dan disimpan kode validasinya ke atribut validasi pada data matakuliah yang divalidasi.

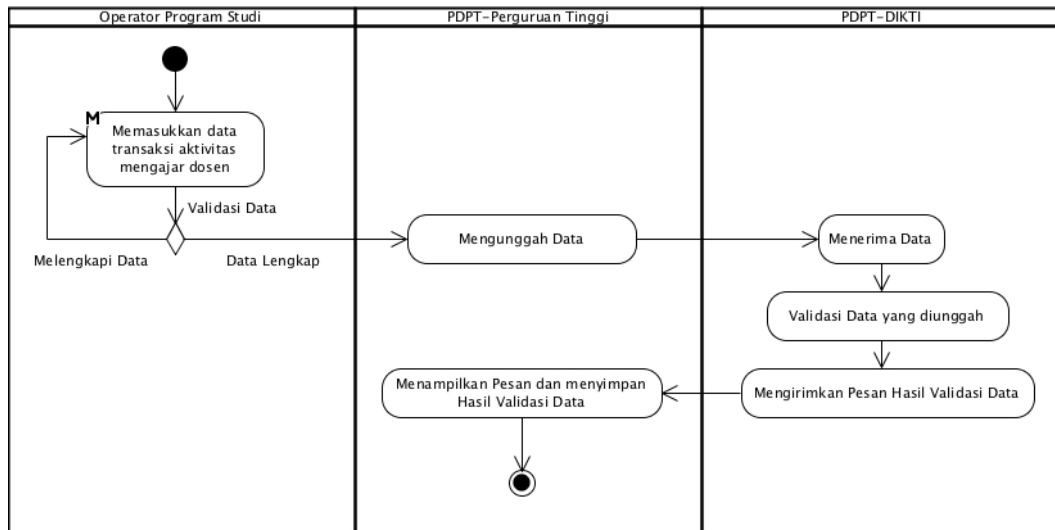
i. Activity diagram pengiriman data transaksi aktivitas kuliah mahasiswa



Gambar 3.10: Activity diagram pengiriman data transaksi aktivitas kuliah mahasiswa

Proses pengiriman data transaksi aktivitas kuliah mahasiswa dimulai dari operator program studi memasukkan data transaksi aktivitas kuliah mahasiswa kedalam system, data yang telah diinput oleh operator program studi dan divalidasi oleh system kemudian dipersiapkan oleh administrator untuk diunggah ke PDPT-DIKTI, setelah administrator mengunggah data ke PDPT-DIKTI, PDPT-DIKTI lalu memvalidasi data yang dikirim oleh system, setelah divalidasi PDPT-DIKTI akan mengirimkan pesan validasi kepada system sebagai bentuk respon dari hasil pengiriman data, pesan tersebut lalu ditampilkan kepada administrator dan disimpan kode validasinya ke atribut validasi pada data transaksi aktivitas kuliah mahasiswa yang divalidasi.

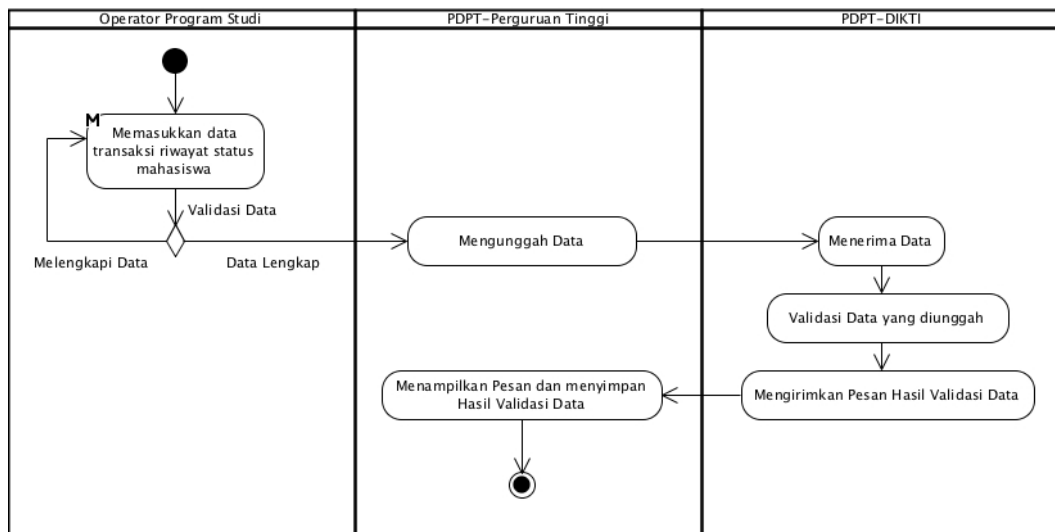
j. Activity Diagram Transaksi Aktivitas Mengajar Dosen



Gambar 3.11: Activity diagram pengiriman data transaksi aktivitas mengajar dosen

Proses pengiriman data transaksi aktivitas mengajar dosen dimulai dari operator program studi memasukkan data transaksi aktivitas mengajar dosen ke dalam sistem, data yang telah diinput oleh operator program studi dan divalidasi oleh sistem kemudian dipersiapkan oleh administrator untuk diunggah ke PDPT-DIKTI, setelah administrator mengunggah data ke PDPT-DIKTI, PDPT-DIKTI lalu memvalidasi data yang dikirim oleh sistem, setelah divalidasi PDPT-DIKTI akan mengirimkan pesan validasi kepada sistem sebagai bentuk respon dari hasil pengiriman data, pesan tersebut lalu ditampilkan kepada administrator dan disimpan kode validasinya ke atribut validasi pada data transaksi aktivitas mengajar dosen yang divalidasi.

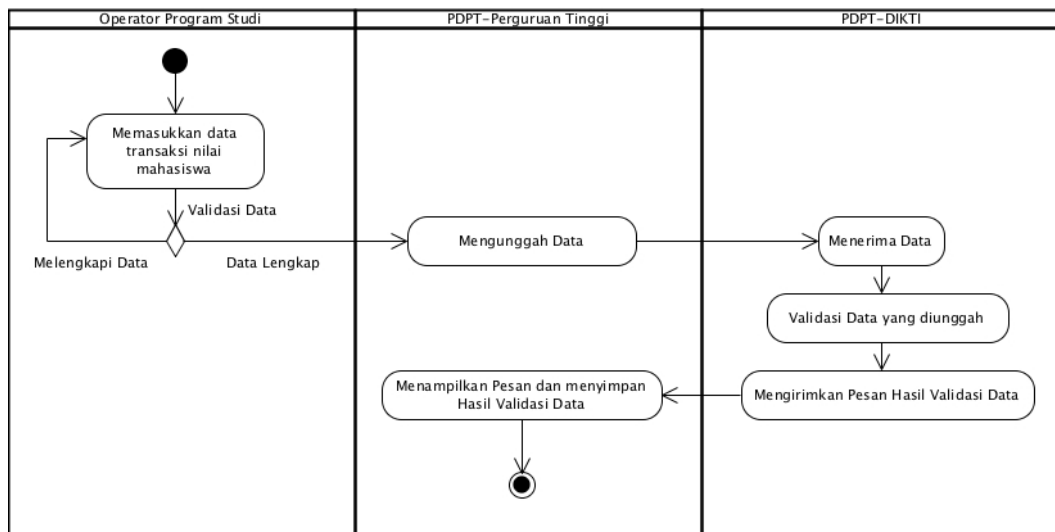
k. Activity Diagram Transaksi Riwayat Status Mahasiswa



Gambar 3.12: Activity diagram pengiriman data transaksi riwayat status mahasiswa

Proses pengiriman data transaksi riwayat status mahasiswa dimulai dari operator program studi memasukkan data transaksi riwayat status mahasiswa kedalam sistem, data yang telah diinput oleh operator program studi dan divalidasi oleh sistem kemudian dipersiapkan oleh administrator untuk diunggah ke PDPT-DIKTI, setelah administrator mengunggah data ke PDPT-DIKTI, PDPT-DIKTI lalu memvalidasi data yang dikirim oleh sistem, setelah divalidasi PDPT-DIKTI akan mengirimkan pesan validasi kepada sistem sebagai bentuk respon dari hasil pengiriman data, pesan tersebut lalu ditampilkan kepada administrator dan disimpan kode validasinya ke atribut validasi pada data transaksi riwayat status mahasiswa yang divalidasi.

I. Activity Diagram Transaksi Nilai Mahasiswa



Gambar 3.13: Activity diagram pengiriman data transaksi nilai mahasiswa

Proses pengiriman data transaksi nilai mahasiswa dimulai dari operator program studi memasukkan data transaksi nilai mahasiswa kedalam sistem, data yang telah diinput oleh operator program studi dan divalidasi oleh sistem kemudian dipersiapkan oleh administrator untuk diunggah ke PDPT-DIKTI, setelah administrator mengunggah data ke PDPT-DIKTI, PDPT-DIKTI lalu memvalidasi data yang dikirim oleh sistem, setelah divalidasi PDPT-DIKTI akan mengirimkan pesan validasi kepada sistem sebagai bentuk respon dari hasil pengiriman data, pesan tersebut lalu ditampilkan kepada administrator dan disimpan kode validasinya ke atribut validasi pada data transaksi nilai mahasiswa yang divalidasi.

D. Jenis Penelitian

Metode yang digunakan dalam penelitian ini adalah metode eksperimental dengan melakukan kajian studi pustaka (*library research*), mengumpulkan data dilapangan (*field research*) dan pembuatan sistem berdasarkan analisis hasil dari pengamatan.

E. Instrumen Pengumpul Data

Instrumen yang digunakan dalam penelitian ini adalah :

1. Perangkat Keras

Perangkat keras yang digunakan adalah satu set Personal Komputer (PC) yang digunakan sebagai standar dalam system komputer

2. Perangkat Lunak

Perangkat Lunak yang digunakan adalah

- a. Sistem operasi mac osx 10.8.4
- b. *Internet Browser* (Mozilla Firefox, Chrome, Safari)
- c. *Integrated Development Environment* (Netbeans, pgAdmin III)

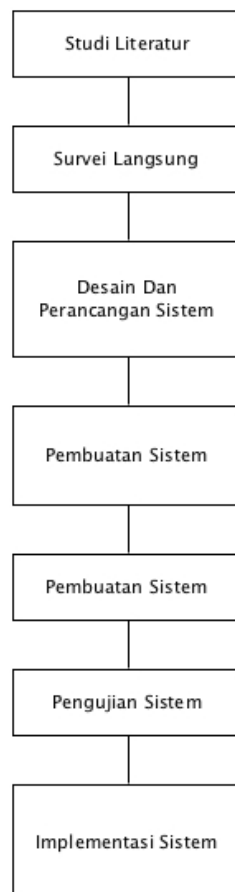
F. Tempat dan Waktu Penelitian

Pengambilan data dan survei pada penelitian ini dilaksanakan pada bulan february tahun 2013, dimana data yang diperoleh berasal dari data pelaporan EPSBED program studi teknik informatika dan program studi

teknik elektro fakultas teknik Universitas Hasanuddin Makassar yang merupakan data pelaporan resmi program studi ke PDPT-DIKTI.

G. Tahapan Penelitian

Tahapan Penelitian Merupakan langkah – langkah yang dilakukan selama proses penelitian sehingga penelitian ini diharapkan dapat lebih terarah. Dalam penelitian ini terdapat enam tahapan yang dilakukan, adapun tahapan tersebut adalah sebagai berikut



Gambar 3.14: Alur Tahapan Penelitian

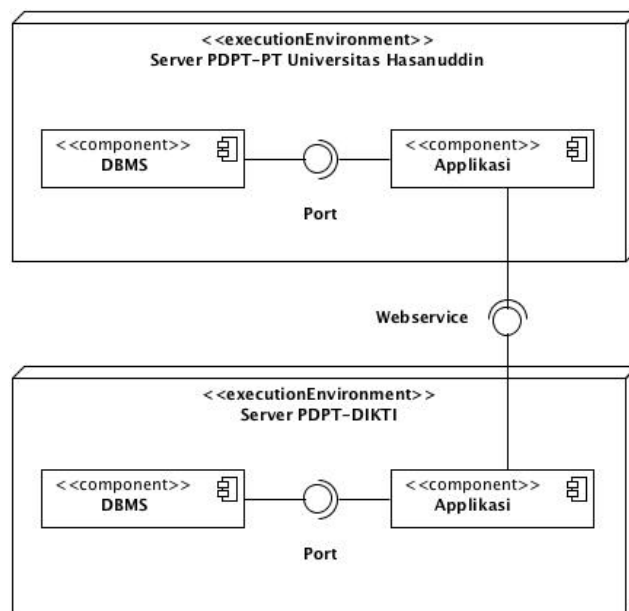
1. melakukan studi literature dengan pengkajian secara teoritis dari berbagai jurnal, artikel ilmiah buku teks yang berhubungan dengan metode dan data yang akan digunakan dalam penelitian ini.
2. Melakukan pemantauan / survey langsung pada program studi teknik elektro S1 dan Teknik Informatika S1 universitas hasanuddin, dimana data yang diperoleh adalah data pelaporan EPSBED yang merupakan data pelaporan dari program studi ke PDPT-DIKTI
3. Desain dan Perancangan Sistem, dimana pada proses ini perancangan system dilakukan dengan menganalisa dokumen – dokumen yang diberikan oleh PDPT-DIKTI melalui halaman website <http://pdpt.dikti.go.id> pada modul panduan
4. Pembuatan Sistem dibangun dengan menggunakan bahasa pemograman PHP versi 5.5 dan menggunakan *database management system* PostgreSQL versi 9.2
5. Pengujian Sistem dilakukan dengan menggunakan metode whitebox dan metode blackbox
6. Implementasi Sistem menggunakan server dengan system operasi Mac OSX 10.8.4, dengan menggunakan *web server* nginx

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. HASIL PENELITIAN

Secara umum, hasil pengembangan system dalam penelitian ini digambarkan melalui diagram berikut :



Gambar 4.1 : Arsitektur pengembangan system antara server PDPT-Universitas dengan PDPT-DIKTI

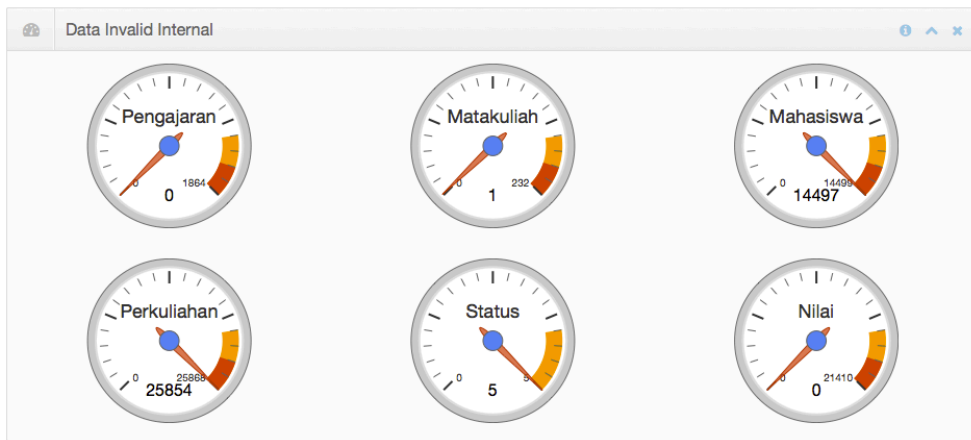
Arsitektur pengembangan sistem terdiri atas *database management system* PostgreSQL dan sistem dimana aplikasi dan DBMS dihubungkan melalui port 5432 dan sistem berada di port 80, data yang dibutuhkan oleh PDPT-DIKTI dikirim melalui layanan web service SOAP yang disediakan oleh PDPT-DIKTI, data yang yang dikirim tersebut kemudian diolah oleh aplikasi dan disimpan ke DBMS PDPT-DIKTI.

1. Sistem PDPT-Universitas

Pada beranda 72system pengguna langsung dihadapkan dengan informasi dalam bentuk chart yang merupakan akumulasi keadaan data pada 72system, terdapat beberapa chart diantaranya adalah :

a. Chart data invalid internal

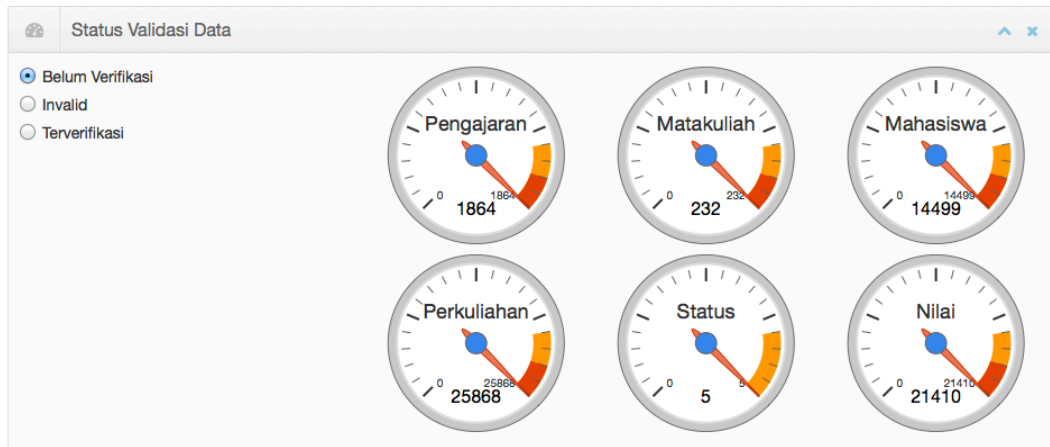
Chart ini menunjukkan keadaan data yang valid atau tidak valid berdasarkan aturan yang digunakan pada kamus data dokumen pendidikan dimana validasi ini bersifat internal dan bukan dari layanan web service dikti, chart tersebut berbentuk gauge yang mewakili enam buah modul yang dibutuhkan oleh PDPT-DIKTI, modul – modul tersebut adalah modul transaksi aktivitas mengajar dosen yang diwakili oleh gauge pengajaran, modul matakuliah yang diwakili oleh gauge matakuliah, modul mahasiswa yang diwakili oleh gauge mahasiswa, modul transaksi aktivitas kuliah mahasiswa yang diwakili oleh gauge perkuliahan, modul transaksi riwayat status mahasiswa yang diwakili oleh gauge status dan transaksi nilai semester mahasiswa diwakili oleh gauge nilai, implementasi dari chart tersebut dapat dilihat pada gambar berikut ini :



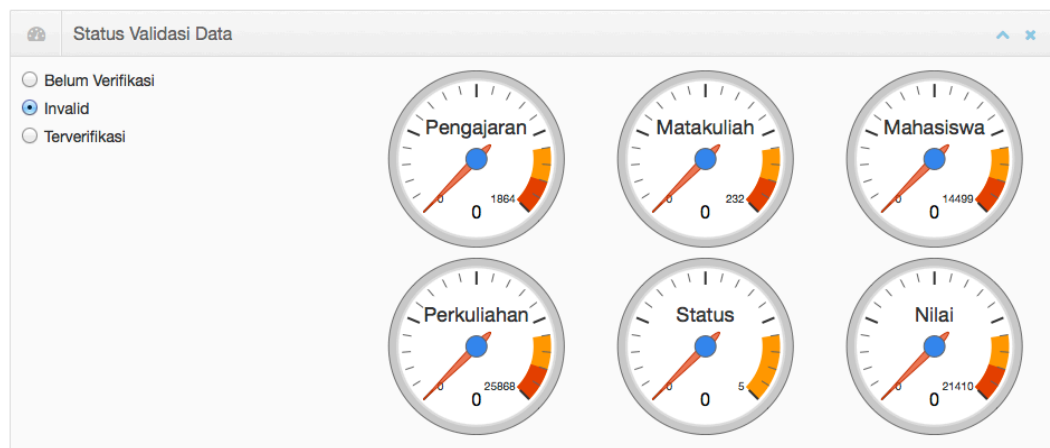
Gambar 4.2 : Chart data invalid internal

b. Chart status validasi data

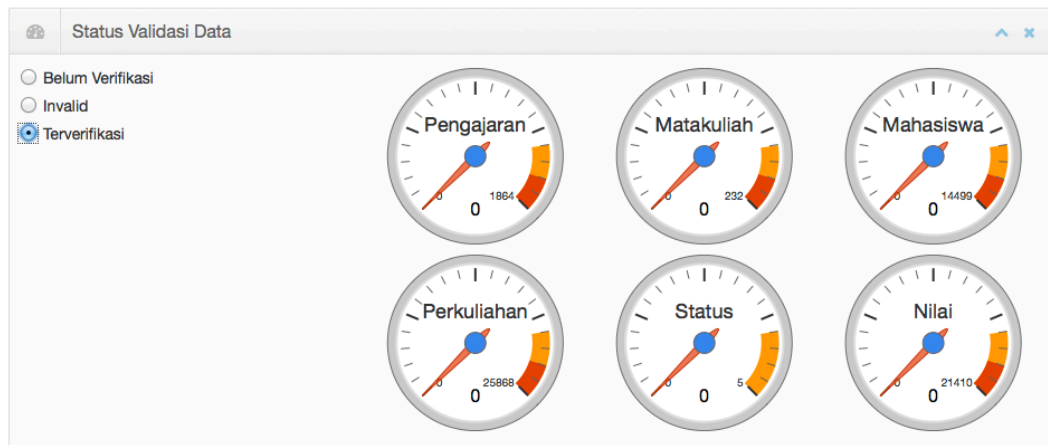
Chart ini berfungsi menunjukkan keadaan status data validasi dari PDPT-DIKTI, enam buah modul data yang dibutuhkan oleh PDPT-DIKTI dimana modul modul tersebut adalah modul transaksi aktivitas mengajar dosen yang diwakili oleh gauge pengajaran, modul matakuliah yang diwakili oleh gauge matakuliah, modul mahasiswa yang diwakili oleh gauge mahasiswa, modul transaksi aktivitas kuliah mahasiswa yang diwakili oleh gauge perkuliahan, modul transaksi riwayat status mahasiswa yang diwakili oleh gauge status dan transaksi nilai semester mahasiswa diwakili oleh gauge nilai, dimana terdapat tiga kategori jenis validasi yaitu belum diverifikasi, invalid, dan terverifikasi, implementasi dari chart tersebut dapat dilihat pada gambar berikut ini :



Gambar 4.3 : chart data yang belum terverifikasi



Gambar 4.4 : chart data yang invalid

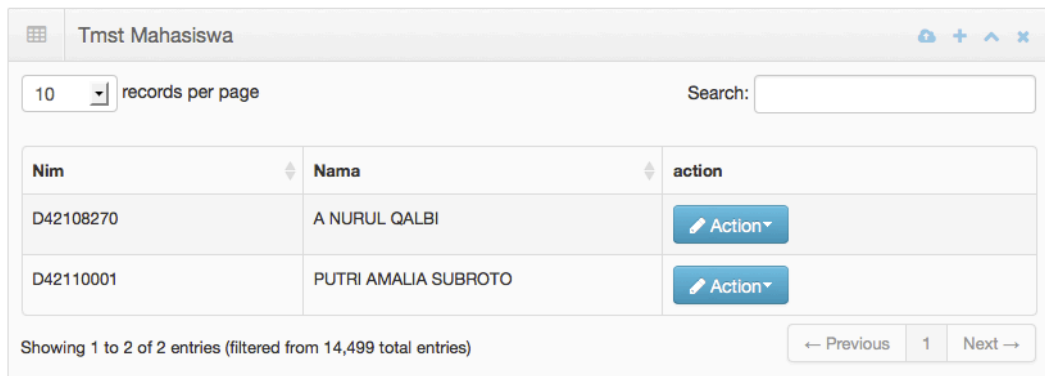


Gambar 4.5 : chart data yang terverifikasi

c. Proses pengiriman data dari 75system ke layanan web service

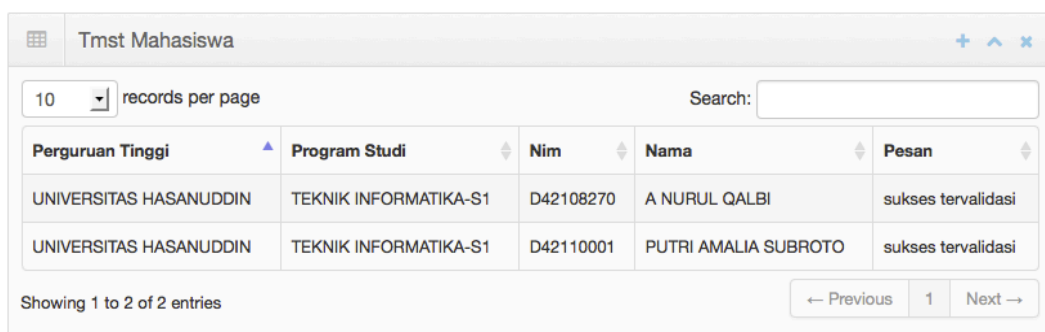
PDPT- DIKTI

proses pengiriman data dari 75system ke layanan webservice dikti dilakukan melalui layanan webservice bertipe soap, dimana data yang akan dikirim disesuaikan dengan dokumen kamus datapendidikan sehingga data yang dikirim dapat diterima sesuai dengan ketentuan PDPT-DIKTI. Dimana dalam implementasinya, pengguna mengirimkan data yang telah validasi oleh 75system melalui modul yang diinginkan, contoh kasus pengguna ingin mengirimkan data mahasiswa yang valid ke layanan web service PDPT-DIKTI, sebelum data dikirimkan 75system akan memperlihatkan data yang akan dikirim ke PDPT-DIKTI seperti yang terlihat pada gambar berikut :



Gambar 4.6 : interface untuk mengirim data Prodi ke PDPT-DIKTI

setelah pengguna menekan tombol upload data yang akan mengirimkan data ke PDPT-DIKTI, setelah data berhasil dikirim maka pengguna akan menerima respon dari layanan web service PTPT-DIKTI apakah data berhasil divalidasi atau data dianggap tidak valid, dimana respon yang diberikan oleh layanan web service dikti dapat dilihat pada gambar berikut ini :



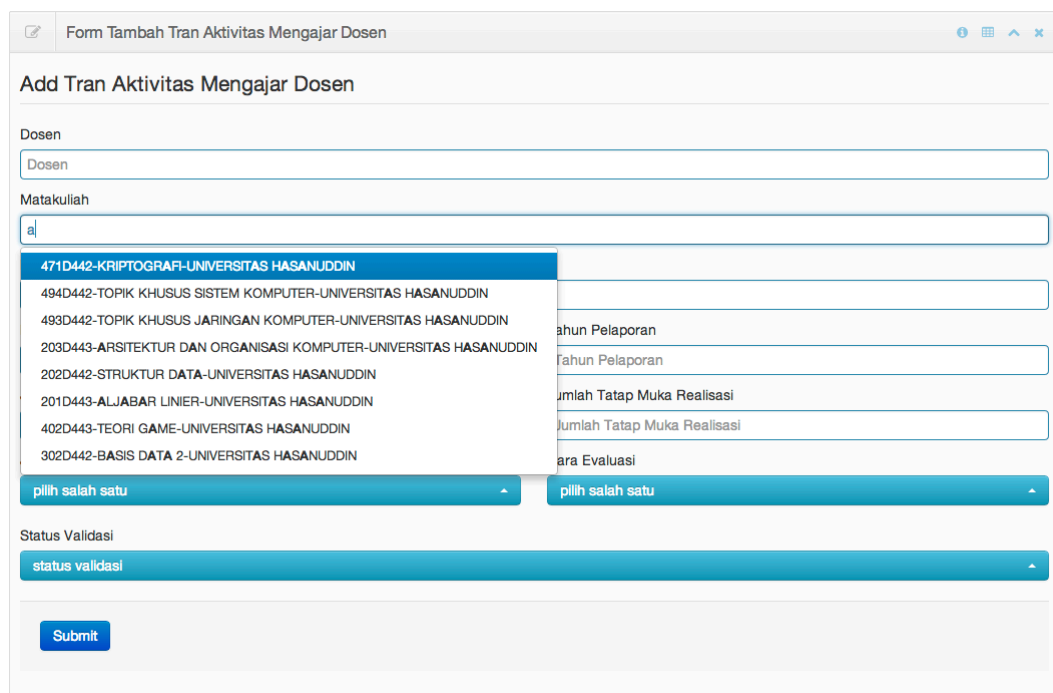
Gambar 4.7 : interface hasil respon data layanan web service PTPT-DIKTI

2. Komponen form sebagai alat bantu pada proses memasukkan data

Komponen form adalah komponen yang digunakan untuk membantu pengguna dalam proses memasukkan data. Dimana beberapa komponen yang digunakan pada form input system yang dibangun antara lain adalah

a. komponen *autocomplete*

komponen autocomplete adalah komponen yang digunakan jika sebuah field form mempunyai banyak sekali pilihan, dimana komponen ini berfungsi memberikan saran inputan kepada pengguna berdasarkan input yang dimasukkan oleh pengguna.



gambar 4.8: komponen autocomplete pada modul transaksi aktivitas mengajar dosen

pada gambar 4.8 dapat dilihat ketika pengguna akan memasukkan data pada field matakuliah maka komponen autocomplete akan memberikan

saran sesuai dengan input yang pengguna masukkan kedalam field matakuliah.

b. Komponen *calendar*

Komponen calendar adalah komponen yang membantu pengguna dalam interaksi pemilihan tanggal.

The screenshot shows a web form titled "Form Tmst Matakuliah" with the following sections and fields:

- Registrasi Matakuliah**
 - Kode: Kode Mata Kuliah
 - Nama: Nama Mata Kuliah
 - Sks Tatap Muka: Sks Tatap Muka
 - Sks Praktikum: Sks Praktikum
 - Sks Praktek Lapangan: Sks Praktek Lapangan
- Program Studi dan Dosen Penanggung Jawab**
 - Perguruan Tinggi: Perguruan Tinggi
 - Dosen: Dosen Pengampuh
 - Tahun Pelaporan: Pilih Tahun Pelaporan
- Tanggal Efektif dan Tahun Pelaporan**
 - Tanggal Mulai Efektif: Tanggal Mulai Efektif
 - Tanggal Akhir Efektif: Tanggal Akhir Efektif (highlighted with a red border)
 - Status Validasi: status validasi
- Bentuk Pelaksanaan**
 - Metode Pelaksanaan Kuliah: Metode Pelaksanaan Kuliah
 - Kelompok Matakuliah: Kelompok Matakuliah
 - Sap: SAP Matakuliah
 - Silabus: Silabus Matakuliah
 - Bahan Ajar: Bahan Ajar
 - Acara Praktek: Acara Praktek
 - Diklat: Diklat Matakuliah
 - Modul Perkuliahan: Modul Matakuliah
 - Kode Kurikulum: kode kurikulum

A calendar widget is displayed over the "Tanggal Akhir Efektif" field, showing the month of July 2013. The date 13 is selected and highlighted in blue.

	Mg	Sn	Sl	Ra	Ka	Ju	Sa
30		1	2	3	4	5	6
7		8	9	10	11	12	13
14		15	16	17	18	19	20
21		22	23	24	25	26	27
28		29	30	31	1	2	3
4		5	6	7	8	9	10

Submit

Gambar 4.9: komponen calendar pada modul matakuliah

Pada gambar 4.9 dapat dilihat komponen calendar membantu interaksi pengguna dengan 79system ketika pengguna akan memasukkan isian pada field tanggal akhir efektif pada modul matakuliah.

3. Validasi Internal Sistem

Validasi dilakukan ketika operator program studi melakukan proses memasukkan data ataupun melakukan proses perubahan data kedalam 79system melalui sebuah form penginputan data, seperti yang terlihat pada contoh form penginputan data pada modul master matakuliah pada gambar berikut

Form Tmst Matakuliah

Registrasi Matakuliah

Kode: Nama:

Sks Tatap Muka: Sks Praktikum: Sks Praktek Lapangan:

Program Studi dan Dosen Penanggung Jawab

Perguruan Tinggi: Dosen: Tahun Pelaporan:

Tanggal Efektif dan Tahun Pelaporan

Tanggal Mulai Efektif: Tanggal Akhir Efektif: Status Validasi:

Bentuk Pelaksanaan

Metode Pelaksanaan Kuliah: Status Mata Kuliah:

Kelompok Matakuliah: Jenis Matakuliah: Kode Kurikulum:

Sap: Silabus: Bahan Ajar:

Acara Praktek: Diktat: Modul Perkuliahan:

Gambar 4.10: Form input data pada modul matakuliah

Dimana pada form tersebut ada form input dengan garis berwarna biru dan form input dengan garis berwarna abu – abu, dimana form input dengan garis berwarna biru adalah form input yang wajib diisi, sedangkan form dengan garis abu – abu adalah data yang bersifat optional untuk diisi.

The image shows a web form for updating course registration. The form is divided into several sections:

- Registrasi Matakuliah:** Includes fields for 'Kode' (071U002), 'Nama' (PENDIDIKAN PANCASILA), 'Sks Tatap Muka' (2), 'Sks Praktikum' (0), and 'Sks Praktek Lapangan' (0).
- Program Studi dan Dosen Penanggung Jawab:** Includes 'Perguruan Tinggi' (UNIVERSITAS HASANUDDIN), 'Dosen' (Dosen Pengampus, highlighted with a red border and a tooltip saying 'Field Wajib Diisi'), and 'Tahun Pelaporan' (2010-1).
- Tanggal Efektif dan Tahun Pelaporan:** Includes 'Tanggal Mulai Efektif' (15-04-2013), 'Tanggal Akhir Efektif' (15-04-2013), and 'Status Validasi' (belum diverifikasi).
- Bentuk Pelaksanaan:** Contains multiple dropdown menus for 'Metode Pelaksanaan Kuliah', 'Status Mata Kuliah', 'Kelompok Matakuliah', 'Jenis Matakuliah', 'Kode Kurikulum', 'Sap', 'Silabus', 'Bahan Ajar', 'Acara Praktek', 'Diktat', and 'Modul Perkuliahan'.

A 'Submit' button is located at the bottom of the form.

Gambar 4.11: Validasi *Client Side* Form input pada modul matakuliah

Pada gambar 4.11 ditunjukkan proses validasi dari pada *client side* dimana system memberikan pesan kepada pengguna bahwa ada field data yang wajib diisi ketika pengguna akan mengirim data ke system. Proses validasi secara *client side* adalah proses validasi yang dilakukan oleh browser sebelum data dikirim ke server system.

Gambar 4.12: Validasi *Server Side* pada form transaksi aktivitas mengajar dosen

Pada gambar 4.12 ditunjukkan proses validasi dari pada *server side* dimana system memberikan pesan kepada pengguna bahwa ada field data yang wajib diisi ketika pengguna sudah mengirim data ke system. Proses validasi secara *server side* adalah proses validasi yang dilakukan oleh system sebelum data disimpan kedalam basis data.

4. Pengujian Sistem dengan metode *whitebox*

pengujian sistem dengan metode *whitebox* bertujuan untuk menguji mekanisme internal dari sebuah berjalan sesuai dengan fungsi yang diinginkan, untuk pengujian dengan metode *whitebox* modul yang diuji adalah modul mahasiswa, dimana dalam pengujian ini menggunakan

framework PHPUnit Testing, dalam implementasinya dapat dilihat melalui kode berikut ini :

```
/**
 * testChangeFromFullFormatToShortFormat method
 *
 * @return void
 */
public function testChangeFromFullFormatToShortFormat() {
    $result = $this->TmstMahasiswa-> changeFromFullFormatToShortFormat
('D42108270-A NURUL QALBI-TEKNIK INFORMATIKA-S1-UNIVERSITAS
HASANUDDIN');
    $expected = 'D42108270-A NURUL QALBI';
    $this->assertEqual($expected, $result);}
}
```

Gambar 4.13 : Skenario Pengujian whitebox

Dimana hasil dari pengujian dapat dilihat pada gambar berikut ini:

```
public function changeFromFullFormatToShortFormat($dataMahasiswa = null) {
    $returnedData = null;
    if (!empty($dataMahasiswa)) {
        $tempDataMahasiswa = explode('-', $dataMahasiswa);
        $countDataMahasiswa = count($tempDataMahasiswa);
        if ($countDataMahasiswa == 5) {
            $returnedData = $tempDataMahasiswa[0] . '-' . $tempDataMahasiswa[1];
        }
    }
    return $returnedData;
}
```

Gambar 4.14 : code hasil dari pengujian whitebox

pada gambar tersebut terlihat bahwa daerah area yang berwarna hijau adalah kode yang dijalankan oleh aplikasi, dari penjalanan testing dengan metode whitebox diperoleh data sebagai berikut :

	Code Coverage			
	Classes and Traits	Functions and Methods	CRAP	Lines
Total	0.00%	52.00%	13 / 25	56.64%
TmstMahasiswa	0.00%	52.00%	13 / 25	56.61%
beforeValidate(\$options = array())		0.00%	0 / 1	0.00%
afterSave(\$created)		0.00%	0 / 1	0.00%
dataTableRequest(\$dataTableParams = array())		0.00%	0 / 1	93.98%
dataTableRequestTmstProgramStudi(\$dataTableParams = array())		0.00%	0 / 1	82.42%
dataTableRequestTrefStatusValidasi(\$dataTableParams = array())		0.00%	0 / 1	82.32%
dataTableRequestTrefTahunPelaporan(\$dataTableParams = array())		0.00%	0 / 1	91.49%
dataTableRequestTrefKelas(\$dataTableParams = array())		0.00%	0 / 1	0.00%
dataTableRequestTrefStatusMahasiswa(\$dataTableParams = array())		0.00%	0 / 1	0.00%
parseSemesterPelaporanToTrefTahunPelaporan()		0.00%	0 / 1	0.00%
getListTmstMahasiswa(\$formatFull = false, \$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
parseDataTmstMahasiswaToId(\$dataMahasiswa = null, \$formatFull = false, \$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
tmstMahasiswaFullToId(\$dataMahasiswa = null)		100.00%	1 / 1	100.00%
tmstMahasiswaToId(\$dataMahasiswa = null, \$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
parseDataTmstMahasiswaToData(\$id = null, \$formatFull = false)		100.00%	1 / 1	100.00%
countInvalidData(\$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
countValidationDataDikti(\$tmstProgramStudiId = null, \$trefStatusValidasiId = null)		100.00%	1 / 1	100.00%
getDataChartTmstMahasiswaTrefKelas(\$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
getDataChartTmstMahasiswaTrefStatusMahasiswa(\$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
getDataChartJenisKeiaminMahasiswa(\$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
getDataChartTahunMasukMahasiswa(\$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
changeFromShortFormatToFullFormat(\$dataMahasiswa = null, \$tmstProgramStudiId = null)		100.00%	1 / 1	100.00%
changeFromFullFormatToShortFormat(\$dataMahasiswa = null)		100.00%	1 / 1	100.00%
dataTableGetValidInternalData(\$dataTableParams = array())		0.00%	0 / 1	0.00%
convertToDataDiktiFormat(\$id = null)		0.00%	0 / 1	0.00%
uploadDataToSimulator()		0.00%	0 / 1	0.00%

Gambar 4.15 : Pengujian modul master mahasiswa dengan metode whitebox

Dari gambar diatas terlihat bahwa area yang berwarna hijau fungsi yang diinginkan sudah memenuhi kriteria yang diinginkan sesuai dengan yang dituliskan pada system unit testing, sedangkan area yang berwarna merah adalah fungsi yang tidak mempunyai scenario pengujian atau belum memenuhi system unit testing.

5. Pengujian Sistem dengan metode *blackbox*

pengujian dengan metode black box adalah pengujian yang tidak memperdulikan mekanisme internal pada sebuah system dan hanya berfokus pada keluaran yang dihasilkan sebagai respon dari pelaksanaan sebuah kondisi yang diinginkan, pada pengujian dengan metode ini

mengambil contoh pada modul mahasiswa, dimana dalam pengujiannya didapatkan hasil sebagai berikut

a. Penggunaan memori metode blackbox

Total penggunaan memori (dalam satuan megabyte) untuk pelaksanaan sebuah request ke system untuk modul mahasiswa menggunakan memori sebanyak 25.57 MB

Peak Memory Use 25.57 MB

Message	Memory use
Component initialization	4.24 MB
Controller action start	7.86 MB
Controller render start	19.42 MB
View render complete	21.72 MB

Gambar 4.16 : Penggunaan memori pada pengujian blackbox

dimana proses – proses yang terjadi adalah proses inialisasi komponen controller memakan memori sebanyak 4.25 MB, proses *action start* sebesar 7.86 MB, proses mulai render tampilan memakan memori sebesar 19.42 MB, dan proses selesai render tampilan sebesar 21.72 MB.

b. Waktu proses metode blackbox

Total waktu proses pengujian (dalam satuan milisecond) untuk pelaksanaan sebuah request ke system untuk modul mahasiswa memakan waktu selama 544ms

Total Request Time: 544 (ms)

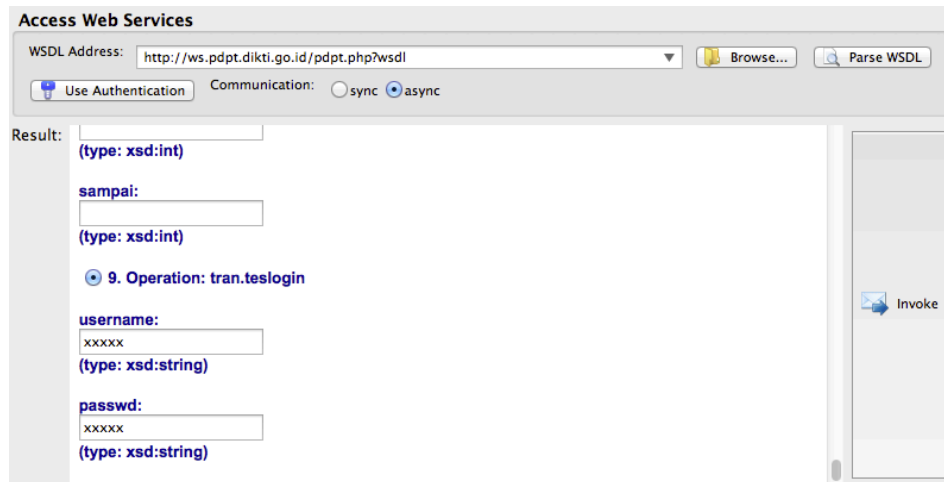
Message	Time in ms	Graph
Core Processing (Derived from S_SERVER["REQUEST_TIME"])	182.03	
Event: Controller.initialize	104.19	
Event: Controller.startup	20.91	
Controller action	116.71	
Event: Controller.beforeRender	32.37	
» Processing toolbar data	32.06	
Rendering View	40.43	
» Event: View.beforeRender	0.14	
» Rendering APP/View/TmstMahasiswa/admin_index.ctp	33.32	
» » Rendering APP/View/Elements/bootstrap/sideBarNavigation.ctp	26.64	
» Event: View.afterRender	0.10	
» Event: View.beforeLayout	0.11	
» Rendering APP/View/Layouts/default.ctp	4.80	
» » Rendering APP/View/Elements/bootstrap/topNavigation.ctp	1.15	
» » Rendering APP/View/Elements/bootstrap/footer.ctp	0.68	
Event: View.afterLayout	0.00	

Gambar 4.17 : Penggunaan memori pada pengujian blackbox

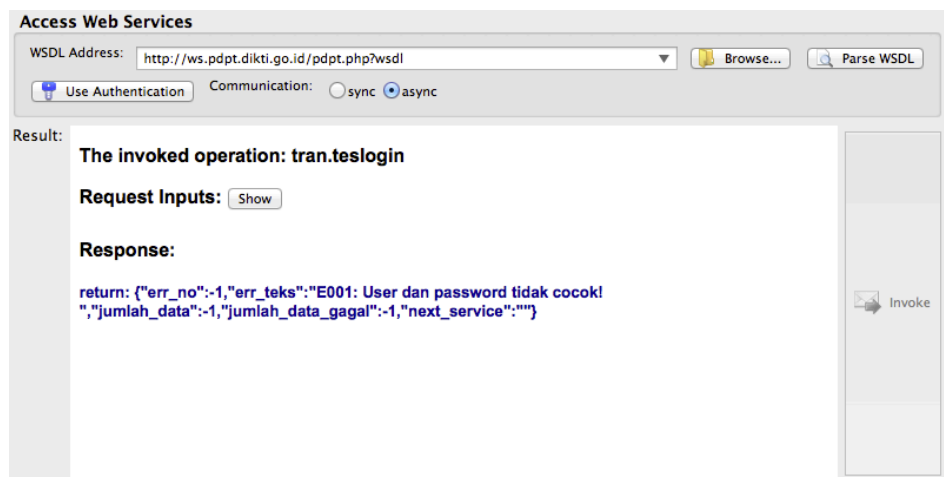
Dimana proses – proses yang terjadi adalah proses *core processing* memakan waktu 182.03ms, proses *event controller initialize* memakan waktu 104.19ms, proses *event controller startup* memakan waktu 20.91ms, proses *controller action* memakan waktu 116.71ms, proses *event controller before render* memakan waktu 32.37ms, proses rendering view memakan waktu 40.43ms.

6. Pengujian fungsional webservice PDPT-DIKTI

Pengujian web service Dikti dilakukan untuk membuktikan bahwa PDPT-DIKTI sebagai penyedia layanan telah siap dengan layanan pengiriman data melalui *web service*, dengan memanfaatkan *plugin* browser firefox, dimana *plugin* ini bertujuan untuk menguji fungsi sebuah layanan *webservice*. Hasil dari pengujian tersebut dapat terlihat dari gambar berikut ini :



Gambar 4.18 : Pengujian Access web service PDPT-DIKTI



Gambar 4.19 : Pengujian Response web service PDPT-DIKTI

Dari hasil percobaan mengakses web service dikti melalui transaksi tes login terlihat jelas bahwa layanan web service PDPT-DIKTI tidak berjalan sesuai dengan apa yang diharapkan, oleh sebab itu untuk menguji apakah system yang dibangun dapat berfungsi mengirimkan data

melalui web service maka disediakan sebuah aplikasi simulator yang dibuat semirip mungkin dengan aplikasi layanan web service PDPT-DIKTI.

7. Simulator PDPT-DIKTI

Simulator PDPT-DIKTI ini dibuat untuk mensimulasikan data yang akan dikirim oleh system ke PDPT-DIKTI dimana simulator ini terdiri atas dua modul yang menerima kiriman data dari system, dimana modul tersebut adalah modul mahasiswa dan modul transaksi aktivitas mengajar dosen, pada gambar berikut terlihat bahwa keadaan simulator pada modul modul mahasiswa dan modul transaksi aktivitas mengajar dosen tidak mempunyai data

Tmst Mahasiswa

Id	Kode Perguruan Tinggi	Kode Program Studi	Nim	Nama Mahasiswa	Actions
Page 1 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0					
<input data-bbox="320 1167 411 1193" type="button" value=" < previous "/> <input data-bbox="421 1167 480 1193" type="button" value=" next > "/>					

Gambar 4.20: Modul Master Mahasiswa pada simulator

Tran Aktivitas Mengajar Dosen

Id	Nidn	Tahun Pelaporan	Semester Pelaporan	Kode Perguruan Tinggi	Kode Program Studi	Kode Jenjang Studi	Kode Matakuliah	Actions
Page 1 of 1, showing 0 records out of 0 total, starting on record 0, ending on 0								
<input data-bbox="314 1487 405 1507" type="button" value=" < previous "/> <input data-bbox="413 1487 443 1507" type="button" value=" next > "/>								

Gambar 4.21: Modul Transaksi Aktivitas Mengajar pada simulator

Setelah data yang dikirim oleh system ke simulator PDPT-DIKTI divalidasi oleh simulator PDPT-DIKTI maka simulator akan menampilkan data yang valid seperti gambar dibawah ini

Tmst Mahasiswa

Id	Kode Perguruan Tinggi	Kode Program Studi	Nim	Nama Mahasiswa	Actions
1	001005	55201	D42110001	PUTRI AMALIA SUBROTO	View Edit Delete
2	001005	55201	D42108270	A NURUL QALBI	View Edit Delete

Page 1 of 1, showing 2 records out of 2 total, starting on record 1, ending on 2

< previous next >

Gambar 4.22: Modul Master Mahasiswa pada simulator setelah memvalidasi data yang dikirim oleh system

Tran Aktivitas Mengajar Dosen

Id	Nidn	Tahun Pelaporan	Semester Pelaporan	Kode Perguruan Tinggi	Kode Program Studi	Kode Jenjang Studi	Kode Matakuliah
1	0027046402	2007	1	001005	20201	3	207D402
2	0020076004	2007	2	001005	20201	3	233D412
3	0026106904	2008	1	001005	20201	3	302D402
4	0001084601	2009	2	001005	20201	3	331D412
5	0011106911	2010	1	001005	26201	3	078U002
6	0005106804	2007	1	001005	26201	3	080U002
7	0001026406	2009	2	001005	20201	3	494D422
8	0017066301	2008	1	001005	22201	3	141D411
9	0001015104	2006	2	001005	20201	3	473D422
10	0011116211	2007	1	001005	20201	3	306D412
11	0027046402	2007	1	001005	20201	3	404D411
12	0020076004	2007	2	001005	20201	3	142D421
13	0003027508	2008	1	001005	20201	3	201D402
14	0011116211	2008	1	001005	20201	3	204D402
15	0019084602	2008	1	001005	20201	3	205D432
16	0031126723	2008	1	001005	20201	3	102D432
17	0009044801	2009	1	001005	20201	3	207D402
18	0023044906	2009	1	001005	20201	3	110D431
19	0027046402	2009	1	001005	20201	3	454D432
20	0001084601	2010	1	001005	20201	3	451D412

Page 1 of 94, showing 20 records out of 1864 total, starting on record 1, ending on 20

< previous 1 2 3 4 5 6 7 8 9 next >

Gambar 4.23: Modul Transaksi Aktivitas Mengajar pada simulator setelah memvalidasi data yang dikirim oleh system

B. PEMBAHASAN

Penggunaan komponen form sebagai alat bantu proses penginputan data pada antarmuka form penginputan data, hal ini dirasa sangat membantu pihak operator dalam melakukan proses penginputan data, dimana operator tidak lagi harus menghafalkan kode, seperti contoh kasus penginputan data transaksi aktivitas kuliah mahasiswa seperti berikut

Form Tambah Tran Aktivitas Kuliah Mahasiswa

Tran Aktivitas Kuliah Mahasiswa

Mahasiswa: Tahun Pelaporan:

D42108001-NUR FADILH GASSING

- D42108112-A M NUR FAJRI
- D42108259-NUR INDRIANI R
- D42108266-RUSMIN NURYADIN
- D42108271-MUH NUR MUSA
- D42109109-NUR JIHAD PUTRI D
- D42109252-NURUL MUHAMMIN R
- D42109255-ASNUR NS

Kelas	Huruf	Angka	Hapus
<input type="text" value="Kode Kelas"/>	<input type="text" value="Bob"/>	<input type="text" value="Nilai"/>	<input type="button" value="Hapus"/>
<input type="text" value="Kode Kelas"/>	<input type="text" value="Bob"/>	<input type="text" value="Nilai"/>	<input type="button" value="Hapus"/>
<input type="text" value="Kode Kelas"/>	<input type="text" value="Bob"/>	<input type="text" value="Nilai"/>	<input type="button" value="Hapus"/>

Gambar 4.24: Proses input pada field mahasiswa pada modul transaksi aktivitas kuliah mahasiswa

Pada gambar 4.24 terlihat pada saat operator ingin menginputkan data pada field mahasiswa pada modul transaksi aktivitas kuliah mahasiswa, dimana system merespon input yang dimasukkan oleh operator dan menampilkan data apa saja yang terkait terhadap input yang dimasukkan oleh operator

Form Tambah Tran Aktivitas Kuliah Mahasiswa

Tran Aktivitas Kuliah Mahasiswa

Mahasiswa: D42109020-FACHRIZAL ALFINULA
Tahun Pelaporan: Tahun Pelaporan

Tran Nilai Semester Mahasiswa

Matakuliah	Kelas	Huruf	Angka	Hapus
471D442-KRIPTOGRAFI	Kode Kelas	Bob	Nilai	⊗
494D442-TOPIK KHUSUS SISTEM KOMPUTER	Kode Kelas	Bob	Nilai	⊗
493D442-TOPIK KHUSUS JARINGAN KOMPUTER	Kode Kelas	Bob	Nilai	⊗

Gambar 4.25: Proses input pada field matakuliah pada modul transaksi aktivitas kuliah mahasiswa

Pada gambar 4.24 terlihat pada saat operator ingin menginputkan data pada field matakuliah pada modul transaksi aktivitas kuliah mahasiswa, dimana sistem merespon input yang dimasukkan oleh operator

operator dan menampilkan data apa saja yang terkait terhadap input yang dimasukkan oleh operator.

Validasi data secara yang dilakukan pada client side maupun server side berfungsi untuk membantu menjaga kualitas data yang akan disimpan dimana form dirancang dengan memberi tanda berupa warna field pada form system, field input dengan garis berwarna biru dan field input dengan garis berwarna abu – abu, dimana field input dengan garis berwarna biru adalah field input yang wajib diisi, sedangkan field dengan garis abu – abu adalah data yang bersifat optional untuk diisi. Hal ini akan membuat operator yang akan melakukan proses penginputan data akan mengetahui field yang wajib diisi dan field yang bersifat optional untuk diisi, validasi pada client side hanya akan memberi tahu pengguna field yang masih belum diisi dengan data ketika pengguna akan mengirim data ke server, sedangkan validasi secara server side akan memvalidasi data isian field wajib diisi atau tidak, field wajib diisi oleh data dengan format tanggal yang benar, field wajib diisi dengan isian angka, ataupun field wajib diisi dengan isian yang bersifat unik. Bentuk validasi ini dapat dilihat pada gambar 4.11 dan gambar 4.12.

Pengujian web service PDPT-DIKTI seperti yang terlihat pada gambar 4.18 dan gambar 4.19, menunjukkan bahwa web service tidak dapat bekerja sebagaimana yang diharapkan, sebagai alternatif pengujian apakah sistem yang dibangun dapat mengirimkan data melalui web service maka dibuat sebuah simulator yang menampung data yang dikirim

oleh system yang dibangun, dimana simulator ini menampung dua modul yaitu modul mahasiswa dan modul transaksi aktivitas mengajar dosen, dimana pada saat proses pengiriman data akan dilakukan system akan menyeleksi data yang valid secara internal, memperlihatkan data tersebut kepada administrator sebelum dilakukan proses pengiriman data seperti yang terlihat pada gambar 4.6, setelah data dikirim terlihat data yang dikirim berhasil disimpan oleh simulator seperti yang terlihat pada gambar 4.22 dan gambar 4.23, sedangkan bentuk respon dari simulator ke system yang dibangun dapat terlihat pada gambar 4.7, dimana system akan memperlihatkan data yang berhasil divalidasi oleh simulator.

Survey system yang dibangun dilakukan secara online , dimana pada survey berhasil mendapat tiga puluh enam responden yang diberikan akses masuk kedalam system yang dibangun, dimana responden yang berada di Indonesia sebanyak tiga puluh satu orang, dari Malaysia sebanyak satu orang, dari Australia sebanyak satu orang dan sebanyak tiga orang berada di Amerika Serikat, dimana ada sepuluh pertanyaan yang diajukan kepada para responden adalah sebagai berikut

Pertanyaan pertama yang diajukan kepada responden adalah

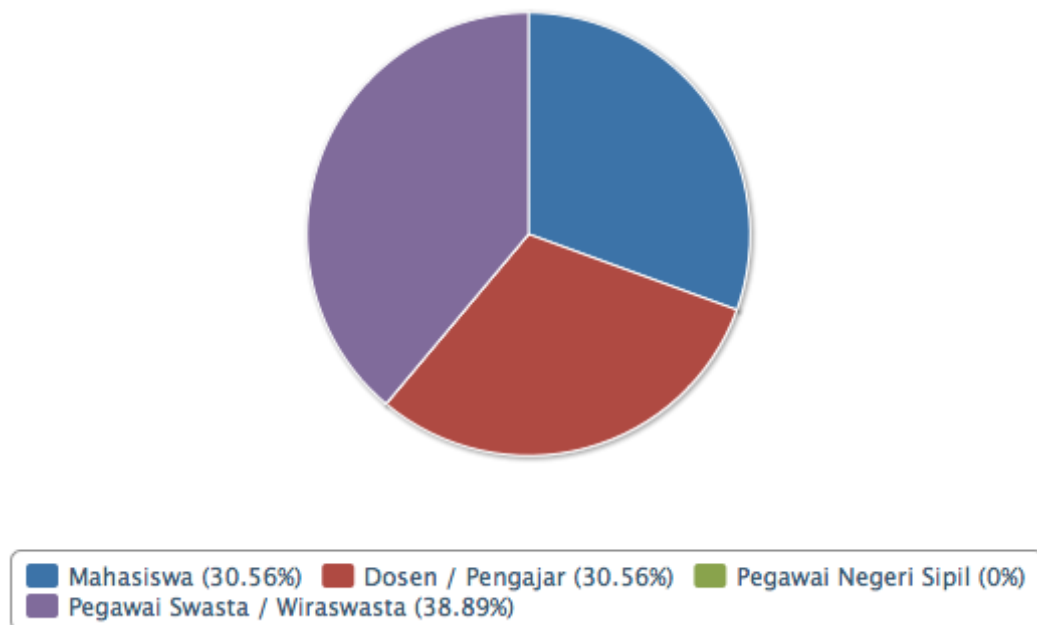
apakah pekerjaan anda saat ?

dengan pilihan jawaban adalah sebagai berikut

- a. Mahasiswa
- b. Dosen / Pengajar
- c. Pegawai Negeri Sipil

d. Pegawai Swasta / Wiraswasta

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.26: hasil survei pertanyaan 1

Dimana responden yang menjawab pekerjaan mereka saat ini adalah mahasiswa sebanyak 30.59%, responden yang menjawab pekerjaan mereka saat ini adalah Dosen atau pengajar sebanyak 30.56%, responden yang menjawab pekerjaan mereka pada saat ini adalah pegawai negeri sipil sebanyak 0%, dan yang menjawab pekerjaan mereka pada saat ini adalah pegawai swasta / wiraswasta sebanyak 38.89%.

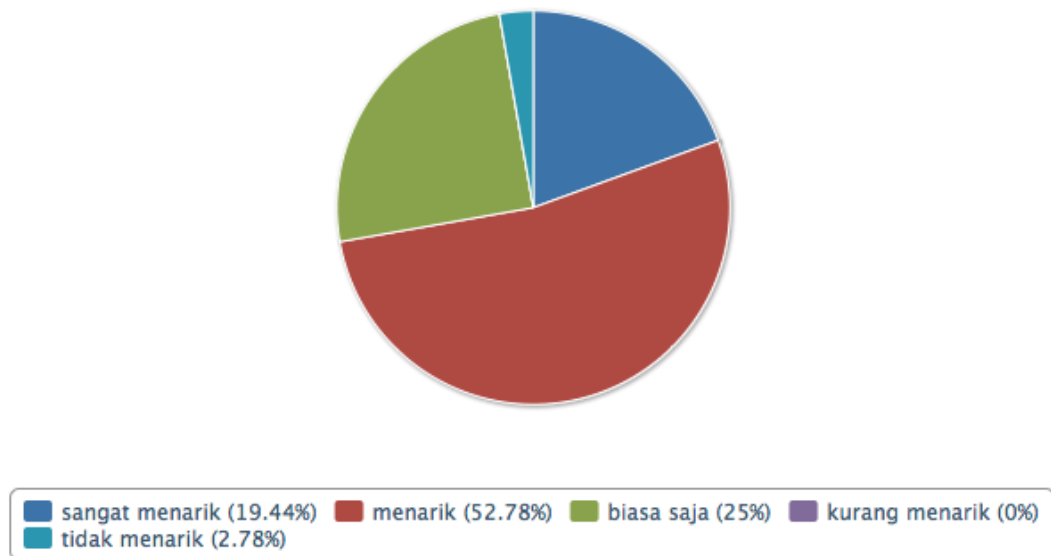
Pertanyaan ke dua yang diajukan kepada responden adalah

Bagaimana menurut anda tampilan antar muka dari aplikasi yang dibangun?

dengan pilihan jawaban adalah sebagai berikut

- a. sangat menarik
- b. menarik
- c. biasa saja
- d. kurang menarik
- e. tidak menarik

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.27: hasil survei pertanyaan 2

dimana responden menjawab bahwa tampilan antar muka dari aplikasi yang dibangun sangat menarik sebanyak 19.44%, responden yang menjawab menarik sebanyak 52.78%, responden yang menjawab biasa

saja sebanyak 25%, responden yang menjawab kurang menarik sebanyak 0% dan responden yang menjawab tidak menarik sebanyak 2.78%.

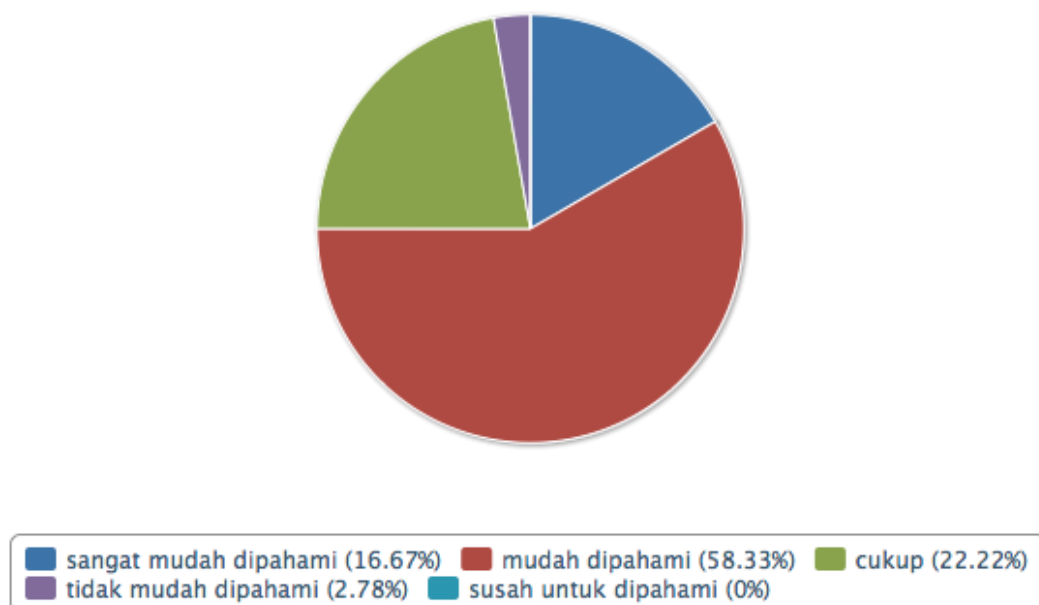
Pertanyaan ke tiga yang diajukan kepada responden adalah

Apakah Struktur tampilan antar muka dari sistem yang dibangun mudah dipahami / dimengerti ?

dengan pilihan jawaban adalah sebagai berikut

- a. sangat mudah dipahami
- b. mudah dipahami
- c. cukup
- d. tidak mudah dipahami
- e. susah untuk dipahami

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.28: hasil survey pertanyaan 3

dimana responden menjawab bahwa struktur tampilan antar muka dari aplikasi yang dibangun sangat mudah dipahami sebanyak 16.67%, responden yang menjawab mudah dipahami sebanyak 58.33%, responden yang menjawab cukup sebanyak 22.22%, responden yang menjawab tidak mudah untuk dipahami sebanyak 2.78% dan responden yang menjawab susah untuk dipahami sebanyak 0%.

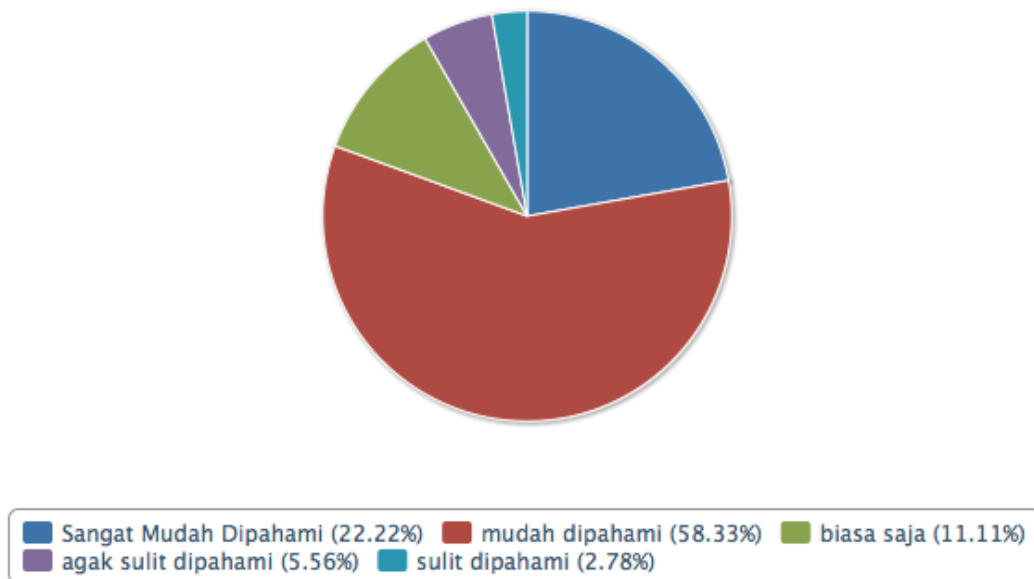
Pertanyaan ke empat yang diajukan kepada responden adalah

Apakah Bahasa yang digunakan pada sistem yang dibangun mudah dipahami ?

dengan pilihan jawaban adalah sebagai berikut

- a. sangat mudah dipahami
- b. mudah dipahami
- c. biasa saja
- d. tidak mudah dipahami
- e. susah untuk dipahami

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.29: hasil survey pertanyaan 4

dimana responden menjawab bahwa Bahasa yang digunakan dari sistem yang dibangun sangat mudah dipahami sebanyak 22.22%, responden yang menjawab mudah dipahami sebanyak 58.33%, responden yang menjawab biasa saja sebanyak 11.11%, responden yang menjawab tidak mudah untuk dipahami sebanyak 5.56% dan responden yang menjawab susah untuk dipahami sebanyak 2.78%.

Pertanyaan ke lima yang diajukan kepada responden adalah

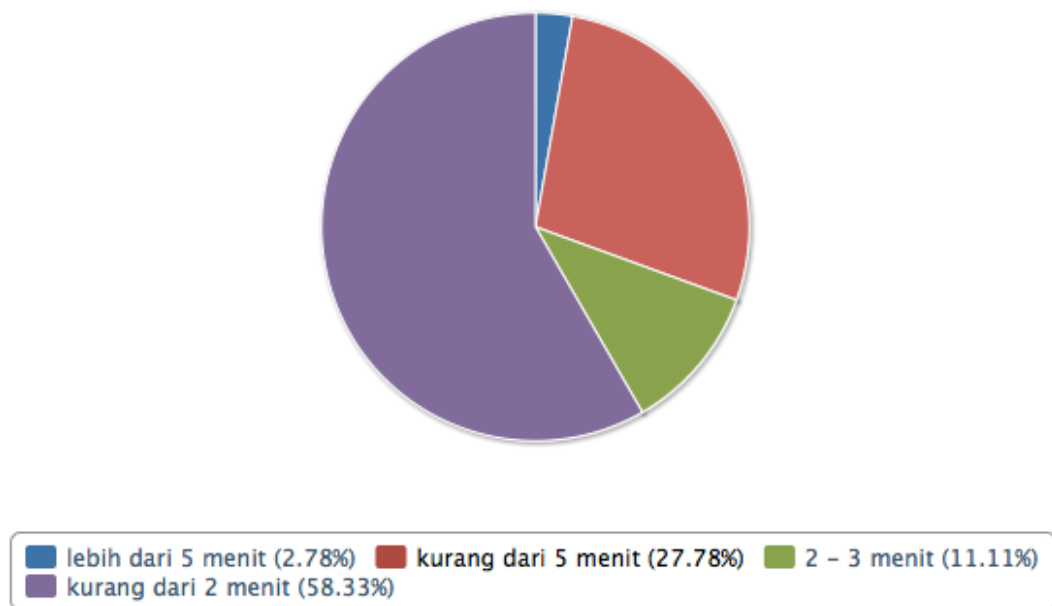
Berapa lama waktu yang anda dibutuhkan untuk mengakses sistem yang dibangun ?

dengan pilihan jawaban adalah sebagai berikut

- a. lebih dari 5 menit
- b. kurang dari 5 menit
- c. 2 – 3 menit

d. kurang dari 2 menit

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.30: hasil survey pertanyaan 5

dimana responden menjawab waktu yang dibutuhkan untuk mengakses sistem yang dibangun lebih dari lima menit sebanyak 2.78%, responden yang menjawab kurang dari lima menit 27.78%, responden yang menjawab dua – tiga menit 11.11%, dan responden yang menjawab kurang dari dua menit sebanyak 58.33%.

Pertanyaan ke enam yang diajukan kepada responden adalah

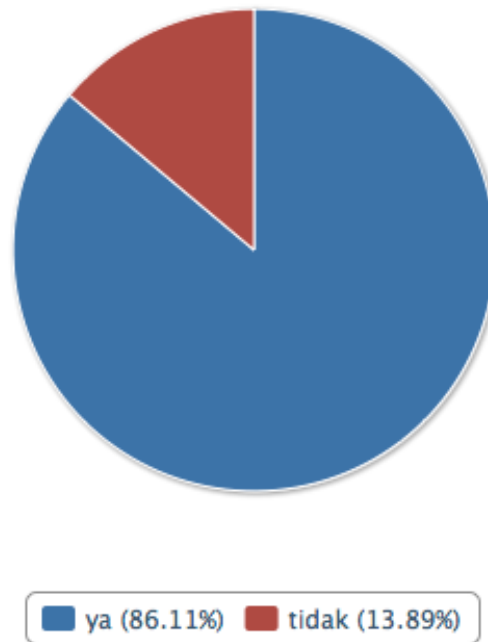
Apakah proses pencarian data menghasilkan data yang anda cari ?

dengan pilihan jawaban adalah sebagai berikut

a. ya

b. tidak

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.31: hasil survey pertanyaan 6

dimana responden menjawab ya pada pertanyaan ke enam sebanyak 86.11%, responden yang menjawab tidak sebanyak 13.89%.

Pertanyaan ke tujuh yang diajukan kepada responden adalah

Apakah proses pencarian data memakan waktu cepat ?

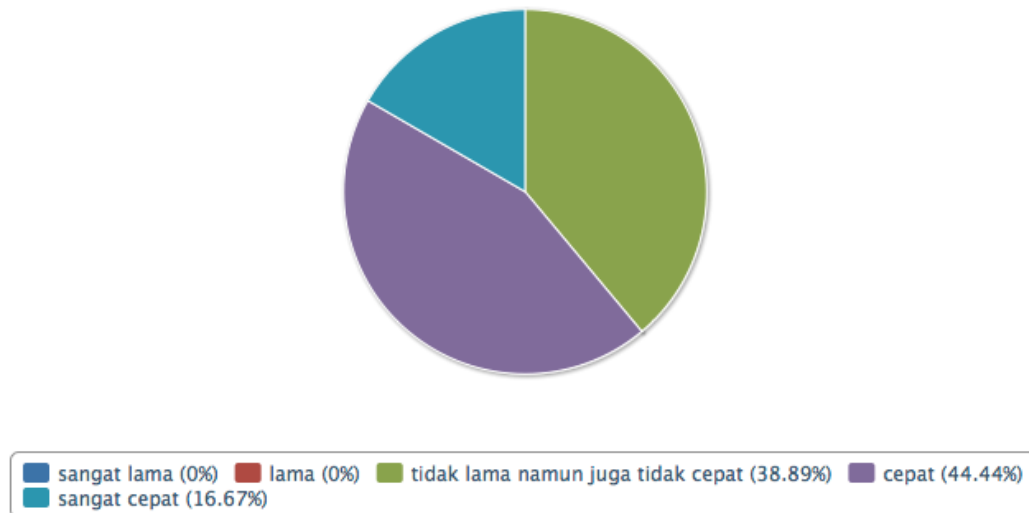
dengan pilihan jawaban adalah sebagai berikut

- a. sangat lama
- b. lama
- c. tidak lama namun juga tidak cepat

d. cepat

e. sangat cepat

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.32: hasil survey pertanyaan 7

dimana responden menjawab bahwa proses pencarian data memakan waktu sangat lama sebanyak 0%, responden yang menjawab lama sebanyak 0%, responden yang menjawab tidak lama namun juga tidak cepat sebanyak 39.89%, responden yang menjawab cepat sebanyak 44.44% dan responden yang menjawab sangat cepat sebanyak 16.67%.

Pertanyaan ke delapan yang diajukan kepada responden adalah

Apakah informasi yang ditampilkan oleh sistem sudah cukup informatif ?

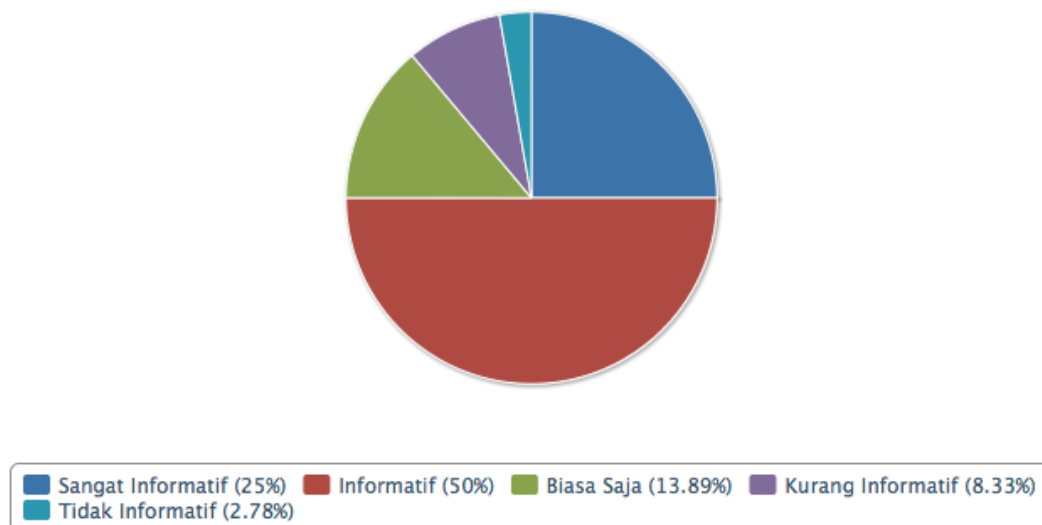
dengan pilihan jawaban adalah sebagai berikut

a. sangat informatif

b. informatif

- c. biasa saja
- d. kurang informatif
- e. tidak informatif

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.33: hasil survey pertanyaan 8

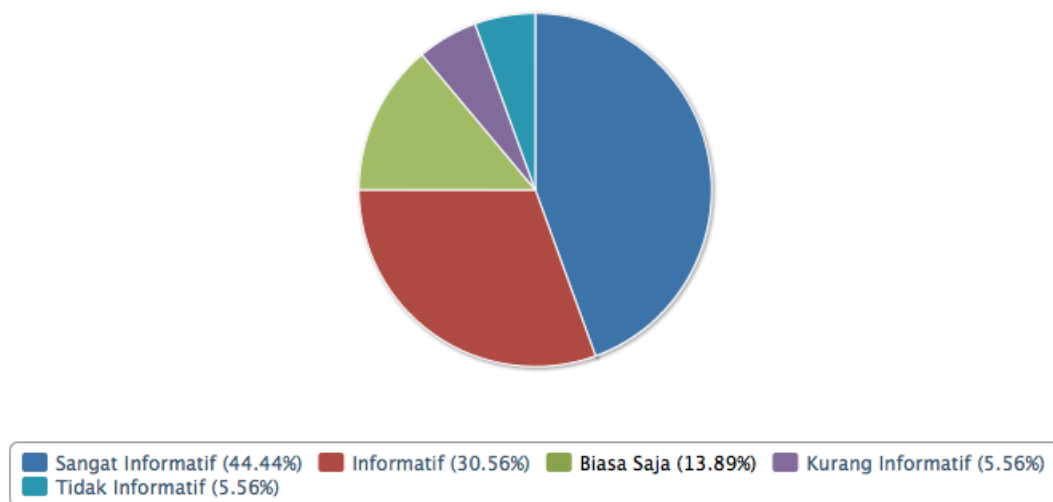
dimana responden menjawab bahwa informasi yang ditampilkan oleh sistem sangat informatif sebanyak 25%, responden yang menjawab informatif sebanyak 50%, responden yang menjawab biasa saja sebanyak 13.89%, responden yang menjawab kurang informatif sebanyak 8.33% dan responden yang menjawab tidak informatif sebanyak 2.78%.

Pertanyaan ke sembilan yang diajukan kepada responden adalah Apakah Chart yang dihasilkan oleh sistem sudah cukup informatif ? dengan pilihan jawaban adalah sebagai berikut

- a. sangat informatif

- b. informatif
- c. biasa saja
- d. kurang informatif
- e. tidak informatif

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.34: hasil survey pertanyaan 9

dimana responden menjawab bahwa Chart yang dihasilkan oleh sistem sangat informatif sebanyak 44.44%, responden yang menjawab informatif sebanyak 30.56%, responden yang menjawab biasa saja sebanyak 13.89%, responden yang menjawab kurang informatif sebanyak 5.56% dan responden yang menjawab tidak informatif sebanyak 5.56%.

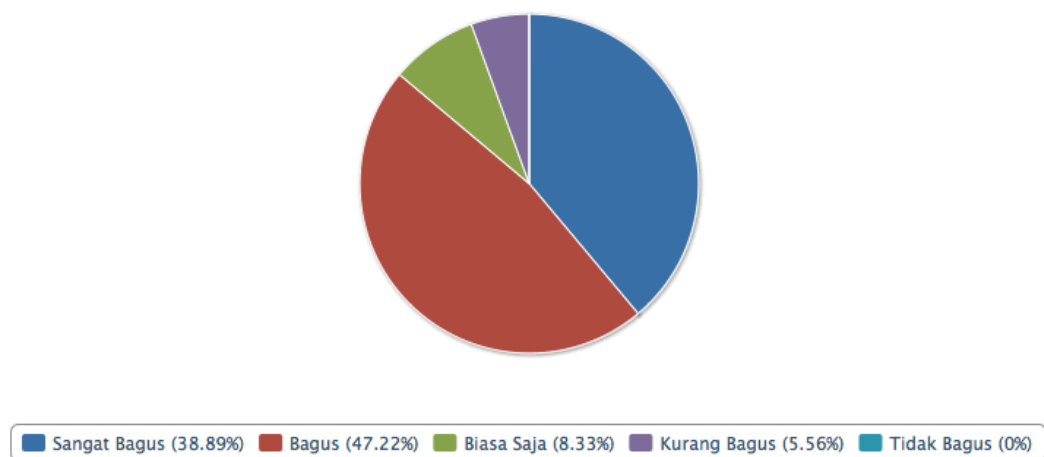
Pertanyaan ke sepuluh yang diajukan kepada responden adalah

Secara Penilaian Keseluruhan dari sistem yang dibangun, bagaimana penilaian anda?

dengan pilihan jawaban adalah sebagai berikut

- a. sangat bagus
- b. bagus
- c. biasa saja
- d. kurang bagus
- e. tidak bagus

Dari jawaban tiga puluh enam responden tersebut diperoleh jawaban seperti dapat dilihat dari chart dibawah ini



Gambar 4.35: hasil survey pertanyaan 10

Responden menilai system yang dibangun secara keseluruhan sangat bagus sebanyak 38.89%, responden yang menjawab bagus sebanyak 47.22%, responden yang menjawab biasa saja sebanyak 8.33%, responden yang menjawab kurang bagus sebanyak 5.56% dan responden yang menjawab tidak bagus sebanyak 0%.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Setelah melakukan tahap – tahap penelitian, mulai dari tahap analisis hingga tahap pengujian, maka dapat diambil kesimpulan pada penelitian ini adalah sebagai berikut :

- a. Sistem yang dibangun bekerja dengan baik hal ini diperkuat dengan hasil survey yang dilakukan.
- b. penggunaan web service sebagai sarana pelaporan data ke PDPT-DIKTI dirasa sangat membantu, karena pelaporan dapat dilakukan sewaktu – waktu tanpa harus menunggu pada periode tertentu dalam pengiriman data.

B. Saran

PDPT-DIKTI sebagai penyedia layanan diharapkan memeriksa kembali ketersediaan layanan web service, agar dapat digunakan sebagai sarana pelaporan data dari perguruan tinggi ke PDPT-DIKTI dan PDPT-DIKTI mengkaji ulang kelengkapan dokumen, baik itu pada dokumen kamus data pendidikan maupun dokumen lain yang menjadi acuan perguruan tinggi untuk membangun sebuah system pelaporan data.

DAFTAR PUSTAKA

1. Anthony David Giordano, 2011, *Data Integration Blueprint and Modeling*, IBM Press, Upper Saddle River New Jersey.
2. David Golding, 2008, *Beginning CakePHP: From Novice to Professional*, Appress.
3. Erich Gamma, Richard Helm, Ralph John, and John Vlissides, 1998, *Design Patterns Element Of Reusable Object Oriented Software*, Addison-Wesley.
4. Jeffrey L.Whitten and Lonney D.Bentley, 2007, *Systems Analysis and Design Methods 7th Edition*, McGraw-Hill.
5. Jennifer Kyrnin, 2013, *Learn HTML CSS XML*, http://webdesign.about.com/od/xml/a/xml_content.htm.
6. M.Tamer Ozsu and Patrick Valduriez, 2011, *Principle of Distributed Database System 3rd edition*, Springer.
7. Mauricio Arregoces and Maurizio Portolani, 2004, *Data Center Fundamentals*, Cisco Systems.
8. Russ Olsen, 2008, *Design Patterns in Ruby*, Addison-Wesley.
9. Ibm developer team, 2013, *webservices*, <https://www.ibm.com/developerworks/webservices/>
10. Cakephp developer team, 2013, *understanding model-view-controller*, <http://book.cakephp.org>
11. Tim PDPT-DIKTI, 2011, *Dokumen Kamus Data Pendidikan*, DIKTI
12. Tim PDPT-DIKTI, 2011, *pemetaan EPSBED dengan evaluasi*, DIKTI
13. Tim PDPT-DIKTI, 2011, *Dokumen Teknis Webservice Push*, DIKTI
14. W3C, <http://www.w3c.org>